

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3

Centro de Gobierno Electrónico



**Aplicación web para la inscripción, revisión técnica y control
de vigencia de disposiciones jurídicas en el MINJUS**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Dailenis Alcántara Frómeta

Tutor: MsC. Yarina Amoroso Fernández

Co-Tutores: Ing. Yordanis García Leiva

Ing. Robin Sencial Terrero

La Habana, 2015.

“Año 57 de la Revolución”

Albert Einstein

Hay una fuerza
motriz más poderosa
que el vapor,
la electricidad y
la energía atómica:
la voluntad



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firman la presente a los _____ días del mes de _____ del año _____.

Firma del autor

Dailenis Alcántara Frómeta.

Firma del tutor

MSc. Yarina Amoroso Fernández

Firma del co-tutor

Ing. Yordanis García Leiva

Firma del co-tutor

Ing. Robin Sencial Terrero

AGRADECIMIENTOS

A mi madre por todo su cariño, comprensión y paciencia, por servirme de guía y apoyo durante toda la vida. Te amo

A mi hermano Enriquito que junto a mi madre es mi sustento, a mi papá, a toda mi familia.

A mi cosito Richard por su apoyo incondicional y darme los momentos más lindos y felices de los últimos dos años.

A la profe Mailén y a mis tutores (Yordanis, Robin y Yarina) por su paciencia y estar en cada momento del desarrollo de este trabajo y que sin su esfuerzo y dedicación no hubiera sido posible. De todo corazón gracias por todo.

A mis amigos en especial a Yade, Yamila, Erlis. Ira por estar ahí cuando los necesite y con los que he pasado muy buenos e inolvidables momentos.

Al tribunal por tantos consejos.

A todos los que de una forma u otra contribuyeron a la realización de este trabajo.

Muchas gracias.

DEDICATORIA

*Le dedico este logro de mi vida mi madre que es mi tesoro máspreciado, por todo su amor,
su sacrificio, y apoyo en las decisiones que he tomado en vida.*

*A mi hermano Enriquito por su ejemplo, los consejos y contribuir en mi formación, por ser mi
amigo.*

RESUMEN

El Ministerio de Justicia de la República de Cuba, es responsable de asesorar al Estado y el Gobierno cubano en materia legislativa, fungiendo como editor y custodio de la Gaceta Oficial de la República de Cuba. En este ministerio se llevan a cabo los procesos de inscripción, revisión técnica y el control de vigencia de las disposiciones jurídicas elaboradas y emitidas por autoridades constitucionalmente reconocidas, las cuales pueden ser: leyes, decretos leyes, decretos, reglamentos, resoluciones, instrucciones y circulares. En la actualidad estos procesos se realizan de forma manual.

La presente investigación se realizó con el objetivo de desarrollar una aplicación web que informatice los procesos antes mencionados. En el trabajo se utilizaron tecnologías libres, cumpliendo con las políticas de soberanía tecnológica establecidas en el país. Como resultado de la investigación, se obtuvo una solución informática que posibilita realizar la inscripción y revisión técnica de los documentos asentados en el Registro de Disposiciones Jurídica del Ministerio de Justicia y controlar la vigencia de los que hayan sido publicados en la Gaceta Oficial, contribuyendo a agilizar el desarrollo de los procesos, ahorrar recursos y certificar la vigencia de las disposiciones jurídicas.

PALABRAS CLAVES: control de vigencia, disposiciones jurídicas, inscripción, Ministerio de Justicia, revisión técnica

ABSTRACT

The Ministry of Justice of the Republic of Cuba, is responsible for advising the Cuban State and Government on legislative subject, serving as editor and custodian of the Official Gazette of the Republic of Cuba. In this ministry are carried out the processes of inscription, technical review and control of validity of the legal provisions elaborated and emitted by constitutionally recognized authorities, which can be: laws, decrees, regulations, resolutions, instructions and circulars. Today these processes are done manually.

The present research was make with the objective to develop a web application that computerized the aforementioned processes. At work free technologies were used, complying with the policies of technological sovereignty established in the country. As a result of the investigation, was obtained a software solution that allows to inscription and technical review of documents settled in the Register of Legal Provisions of the Ministry of Justice. As a result of the research, was obtained a software solution that allows to inscription and technical review of documents settled in the Register of Legal Provisions of the Ministry of Justice and control the observance of which have been published in the Official Gazette. All this contributes to accelerate the development of processes, save resources and certify the validity of the legal provisions.

KEYWORDS: *control of validity legal provisions, inscription, Ministry of Justice, technical review*

ÍNDICE DE CONTENIDOS

Introducción.....	12
Capítulo 1: Fundamentación teórica de la investigación.....	16
1.1 Introducción.....	16
1.2 Marco conceptual.....	16
1.2.1 Registro de Disposiciones Jurídicas.....	16
1.2.2 Publicidad de Derecho.....	16
1.2.3 Gaceta Oficial.....	17
1.2.4 Publicidad electrónica.....	18
1.3 Tendencias actuales.....	18
1.4 Metodología de desarrollo de software.....	19
1.4.1 Extreme Programming (XP).....	21
1.5 Herramienta CASE.....	22
1.6 Lenguaje de Modelado Unificado (UML).....	23
1.7 Sistemas Gestor de contenido (CMS).....	23
1.7.1 WordPress.....	24
1.7.2 Joomla.....	24
1.7.3 Drupal.....	25
1.7.4 Selección del sistema de gestión de contenido a utilizar.....	26
1.8 Lenguajes de programación.....	26
1.8.2 Lenguaje programación del lado del servidor.....	26
1.8.3 Lenguajes de programación del lado del cliente.....	27
1.9 Sistemas Gestores de Base de Datos.....	28
1.9.2 PostgreSQL.....	28
1.9.3 MySQL.....	29
1.9.4 SQLite.....	30
1.9.5 Selección del sistema de gestión de base de datos a utilizar.....	30
1.10 Entorno de desarrollo integrado.....	31
1.10.1 Netbeans IDE.....	31
1.11 Servidor web.....	31
1.11.1 Apache Server.....	32
1.12 Conclusiones parciales.....	32
Capítulo 2: Análisis y diseño de la aplicación web.....	33
2.1 Introducción.....	33
2.2 Descripción de la aplicación web.....	33

2.3	Fases del proceso de desarrollo	35
2.3.1	Exploración	35
2.3.1.1	Requisitos del sistema	35
2.3.1.2	Historias de usuarios.....	38
2.3.2	Planificación de la entrega.	40
2.3.2.1	Estimación de esfuerzo por historias de usuarios	40
2.3.3	Iteraciones	41
2.3.3.1	Plan de duración de las iteraciones.....	41
2.3.4	Arquitectura del sistema	42
2.4	Diseño de la aplicación web	44
2.4.1	Tarjetas CRC (Clase, Responsabilidad y Colaboración).....	44
2.4	Patrones de diseño.....	45
2.4.1	Patrones de diseño de base de datos.....	45
2.4.2	Patrones implementados en el CMS Drupal.	46
2.4.3	Otros patrones de diseño empleados.	47
2.5	Descripción de la base de datos	48
2.6	Conclusiones parciales.....	49
Capítulo 3: Implementación y pruebas de la aplicación web.....		50
3.1	Introducción	50
3.2	Implementación.....	50
3.2.1.	Tareas de ingeniería	50
3.2.2	Estándares de codificación	51
3.2	Diagrama de despliegue.....	52
3.3	Técnicas de validación de los requisitos.....	53
3.4	Validación del diseño	53
3.5	Pruebas	56
3.5.1	Pruebas unitarias.....	56
3.5.2	Pruebas de aceptación	61
3.6	Aporte de la investigación	62
3.7	Conclusiones parciales.....	63
Conclusiones Generales		64
Recomendaciones.....		65
Bibliografía Referenciada		66
Bibliografía Consultada		69
Anexos		72
Anexo 1: Acta de Aceptación de la revisión de los requisitos.		72
Anexo 2: Acta de Liberación Interna de Productos de Software.		73

ÍNDICE DE CONTENIDOS

Anexo 3: Acta de aceptación de la aplicación web por parte del cliente.	74
Anexo 4: Modelo de datos.	75

ÍNDICE DE FIGURAS

Figura 1: Vista general de la aplicación web.....34

Figura 2: Diagrama de componentes.....43

Figura 3: Empleo del patrón llaves subrogadas en la entidad Documento.....45

Figura 4: Clase experta en información referente a materias de las disposiciones jurídicas. 47

Figura 5: Presencia del patrón Alta Cohesión en la aplicación.48

Figura 6: Modelo de datos.....49

Figura 7: Diagrama de despliegue.....52

Figura 8: Representación en (%) de los resultados de la aplicación de la métrica TOC.54

Figura 9: Representación en (%) de los resultados de la aplicación de la métrica RC.....56

Figura 10: Función Automatizado_cron.57

Figura 11: Grafo del camino básico de la función Automatizado_cron.....58

Figura 12: Resultados obtenidos de la aplicación de la técnica del camino básico.....60

Figura 13: No conformidades detectadas en la aplicación a través de las pruebas de caja negra.....61

Figura 14: Acta de aceptación de los requisitos72

Figura 15: Acta de liberación de la aplicación web73

Figura 16: Acta de aceptación de la aplicación web74

Figura 17: Modelo de datos.....75

ÍNDICE DE TABLAS

Tabla 1: Metodologías ágiles más usadas (Highsmith, 2002).	20
Tabla 2: Historia de Usuario: Gestionar disposición jurídica en el RGDJ.	39
Tabla 3: Historia de Usuario: Control de vigencias de las disposiciones.	39
Tabla 4: Estimación de esfuerzo por Historia de Usuario	40
Tabla 5: Plan de duración de las iteraciones.	41
Tabla 6: TarjetaCRC_Automatizado.	44
Tabla 7: Tarea de ingeniería 2 Control de vigencia de disposiciones jurídicas.	50
Tabla 8. Rango de valores para medir la afectación de los atributos de calidad (TOC).	54
Tabla 9. Rango de valores para medir la afectación de los atributos de calidad (RC).	55
Tabla 10: Caso de prueba de caja blanca para el camino básico #2.	59
Tabla 11: Caso de prueba de caja negra de la funcionalidad Registrar disposición jurídica.	60
Tabla 12: Caso de prueba de Aceptación de la HU “Control de vigencia de la disposición jurídica”.	62

Introducción

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) tiene un protagonismo cada vez mayor, para la automatización de los procesos en las instituciones. Los avances tecnológicos han producido cambios en el sector editorial contemporáneo, transformando el formato impreso al digital. El surgimiento de nuevas publicaciones y el uso de las telecomunicaciones para su distribución, han posibilitado un mayor crecimiento en las publicaciones electrónicas. Cuba no está exenta de estos avances y ha ido introduciendo paulatinamente estas tecnologías en diversas ramas, tal es el caso del Ministerio de Justicia de la República de Cuba (MINJUS). Esta institución forma parte de la Administración Pública cubana, se encarga de asesorar al Estado y el Gobierno en materia legislativa y funge como editor y custodio de la Gaceta Oficial de la República de Cuba (GO), órgano oficial de publicidad normativa en el país, que cuenta con una página Web.

Asociado a esta función de editor y asesor en materia de legislación, en el MINJUS se desarrollan los procesos de inscripción de las disposiciones jurídicas que elaboran quienes tienen, en virtud de la Constitución, la facultad para disponerlas. Esta institución también realiza la revisión técnica y el control de vigencia de las mismas, cuando son publicadas en la GO, esta última función constituye una facultad de certificación.

En Cuba, a partir de la vigencia de la Constitución de 1976, las disposiciones jurídicas pueden ser: leyes, decretos leyes, decretos, reglamentos, resoluciones, instrucciones y circulares, entendiéndose como rangos normativos que pueden adoptar las disposiciones legales emitidas por las autoridades constitucionalmente reconocidas. Los emisores oficiales de disposiciones jurídicas remiten al MINJUS los actos normativos para su evaluación e inclusión en la Gaceta Oficial de la República de Cuba (GO), órgano oficial de publicidad normativa en el país.

Los actos normativos enviados al MINJUS son recepcionados en la Dirección de Legislación y Asesoría e inscritos en el Registro General de Disposiciones Jurídicas. A partir de la inscripción en este registro, se desarrolla el procedimiento de evaluación técnica que realiza este ministerio como máximo órgano de asesoría del Estado y el Gobierno en materia legislativa.

En la actualidad las disposiciones jurídicas llegan al MINJUS en soporte papel y el proceso de inscripción y evaluación técnica de las mismas se continúa haciendo de forma manual. Esto provoca lentitud en el desarrollo del proceso, al tardar más de 15 días entre la fecha de salida de la disposición del organismo emisor hasta ser revisada por los especialistas del

MINJUS y pasar a la edición para su publicación en la GO. Además se generan grandes cantidades de documentos, que por el volumen se tornan difíciles de gestionar, imposibilitando la lectura simultánea de una misma disposición jurídica por más de un especialista.

Por otra parte el MINJUS no cuenta con un mecanismo que permita de forma rápida certificar la vigencia en base a la publicación de la norma en la GO, pues la página Web no brinda este servicio y para emitir esta certificación hay que acudir al archivo físico.

Otras de las deficiencias que caracterizan el proceso de inscripción, revisión técnica y control de vigencia de las disposiciones jurídicas, es el gasto de recursos, pues los organismos emisores tienen que imprimir sus disposiciones para poder personarse en el MINJUS y entregar las mismas.

La situación problemática anteriormente descrita permitió identificar como **problema de la investigación**:

¿Cómo realizar la inscripción y revisión técnica de los documentos asentados en el Registro de Disposiciones Jurídica del MINJUS y controlar la vigencia de los que hayan sido publicados en la GO contribuyendo a agilizar el desarrollo del proceso, ahorrar recursos y certificar la vigencia de las disposiciones jurídicas?

Para solucionar el problema se define como **objeto de estudio**: Aplicaciones web para la inscripción, revisión y control de vigencia de documentos

Determinándose como **objetivo general**: Desarrollar una aplicación web que permita realizar la inscripción y revisión técnica de los documentos asentados en el Registro de Disposiciones Jurídica del MINJUS y controlar la vigencia de los que hayan sido publicados en la GO contribuyendo a agilizar el desarrollo del proceso, ahorrar recursos y certificar la vigencia de las disposiciones jurídicas. Declarándose los **objetivos específicos** siguientes:

- Elaborar el marco teórico de la investigación en función de los términos relacionados con los procesos de inscripción, evaluación y control de disposiciones jurídicas.
- Desarrollar una aplicación web que permita la inscripción, evaluación y control de disposiciones jurídicas.
- Realizar pruebas de validación a la aplicación web propuesta, aplicando técnicas y métricas de validación de software.

Para ello se identifica como **campo de acción**: Aplicaciones web para la inscripción, revisión y control de vigencia de disposiciones jurídicas.

Definiéndose como **idea a defender**: El desarrollo de una aplicación web que permita realizar la inscripción y revisión técnica de los documentos asentados en el Registro de Disposiciones Jurídica del MINJUS y controlar la vigencia de los que hayan sido publicados en la GO contribuirá a agilizar el desarrollo del proceso, ahorrar recursos y certificar la vigencia de las disposiciones jurídicas.

El presente estudio se desarrolló con el empleo de los **métodos científicos de investigación** que se describen a continuación.

Métodos teóricos:

Histórico-Lógico: Se utilizó para identificar las tendencias actuales en relación al desarrollo de sistemas de gestión de documentación legal, permitiendo conocer y estudiar las soluciones más importantes realizados hasta el momento, vinculados con el tema de investigación.

Analítico-Sintético: Se utiliza para analizar la documentación relacionada con los procesos de consulta y control de vigencia de disposiciones legales, organizarla y sintetizarla para lograr una estructura adecuada que permita desarrollar una correcta investigación. También permitió estudiar e identificar la metodología, herramientas y lenguajes más apropiados para el desarrollo de la solución propuesta.

Modelación: Posibilitó la obtención de los principales artefactos generados en cada una de las fases del proceso de desarrollo de software.

Método empírico:

Entrevista: Se realiza para obtener información que permita entender mejor los procesos, identificar las necesidades del cliente y los requisitos que debe cumplir el sistema una vez finalizado este.

Medición: Este se evidencia con las pruebas de validación que se realizan a la aplicación web.

Estructura de la investigación

La presente investigación está estructurada en los tres capítulos que se describen a continuación:

Capítulo 1: Fundamentación teórica de la investigación: Se describen conceptos relacionados con la problemática antes mencionada, así como el estudio de las características fundamentales de los sistemas de gestión de documentación legal existentes en la actualidad para la inscripción, revisión y control de vigencia de esta. Además se describe la metodología, tecnologías y herramientas utilizadas en el desarrollo del trabajo.

Capítulo 2: Análisis y diseño de la aplicación web. Se realiza una descripción de las principales características de la aplicación web, así como el diseño de la misma, partiendo del estudio del estado del arte realizado en el capítulo anterior y teniendo en cuenta las fases definidas por la metodología XP (Extreme Programming o Programación Extrema). Lo anterior se realiza centrándose fundamentalmente en los requisitos del software descritos a través de las Historias de Usuario. Además se analizan elementos importantes dentro del proceso de desarrollo tales como: plan de iteraciones, arquitectura, patrones de diseño y tarjetas CRC (Clase, Responsabilidad y Colaborador).

Capítulo 3: Implementación y pruebas de la aplicación web. Se da continuidad a las fases de la metodología XP, con la implementación y pruebas a la aplicación web, describiéndose los principales artefactos generados, tales como: las tareas ingenieriles por cada historia de usuario, las pruebas y métricas empleadas en la validación.

Para finalizar se presentan las conclusiones, recomendaciones, referencias bibliográficas, bibliografía consultada y los anexos para una mejor comprensión de la investigación.

Capítulo 1: Fundamentación teórica de la investigación.

1.1 Introducción

En este capítulo se define el fundamento teórico de la investigación, inicialmente se precisan algunos conceptos acerca del tema en cuestión para una mejor comprensión del mismo, así como el análisis de experiencias desarrolladas en otras instituciones internacionales. Además se presenta una breve descripción sobre las características fundamentales de la metodología, herramientas y tecnologías seleccionadas para el desarrollo de la aplicación web.

1.2 Marco conceptual

Para una mejor comprensión del tema de la investigación se puntualizan algunos conceptos relacionados con la gestión de información en materia legislativa.

1.2.1 Registro de Disposiciones Jurídicas

El Registro de Disposiciones Jurídicas (RGDJ) constituye el sistema de inscripción y certificación de la vigencia de las disposiciones jurídicas emitidas por los organismos constitucionalmente facultados para ello. En virtud de la inscripción en el Registro se procede al análisis técnico de la disposición y se realiza el proceso de edición y publicación en la Gaceta Oficial de la República de Cuba. Corresponde al Registro General de Disposiciones Jurídicas emitir la certificación de vigencia de las disposiciones jurídicas en Cuba (Diccionario de Jurisprudencia, 2000).

1.2.2 Publicidad de Derecho

Conocer el Derecho constituye un derecho ciudadano y la vez se erige en una responsabilidad del Estado de darlo a conocer, es ello una de las razones de que en los textos constitucionales se establezcan preceptos que establecen las normas fundamentales que rigen la Publicidad de las normas (Amoroso, 2002).

Las disposiciones jurídicas tales como leyes, decretos, decretos leyes, reglamentos, resoluciones, instrucciones y circulares, se publican en la GO con previa inscripción en el Registro de Disposiciones Jurídicas a cargo del Ministerio de Justicia de la República de Cuba.

El conjunto de disposiciones normativas aprobadas y publicadas oficialmente por el Estado constituyen el orden jurídico y precisan alcanzar un carácter sistémico dentro del cual se expresen de manera coherente los aspectos relacionados con la vigencia, la validez y eficacia

de las normas jurídicas. En tal sentido, es necesario que se generen los procedimientos sistemáticos y permanentes del sistema en funciones para alcanzar tales fines en el contexto de la vida social. Un elemento para ello lo constituye el Registro de Disposiciones Jurídicas en lo relativo a la revisión técnica que realiza la Dirección de Legislación del MINJUS, dónde se materializa la función asesora en materia de Derecho al Estado y el Gobierno (Amoroso, 2002).

La fecha de publicación de las disposiciones jurídicas en la GO constituye una condición para fijar la vigencia de las mismas, pero no determina el inicio de la entrada en vigor de la ley y por ello no se hace obligatoria su observancia, cuando la propia ley aplaza el comienzo de su entrada en vigor. (Amoroso, 2015)

Por tal motivo, luego de su publicación en la GO como órgano de publicidad normativa, es necesario llevar un control de las vigencias de las disposiciones legales que permita al usuario obtener resultados más certeros en su consulta. Dicho proceso de consulta y control de vigencias de las disposiciones jurídicas tiene alusión en la Constitución de la República de Cuba, en el artículo 77, que expresa: " Las leyes aprobadas en la Asamblea Nacional del Poder Popular entran en vigor en cada caso determine la propia ley" (Constitución de la República de Cuba, 2013) .De manera que aquí se genera un vínculo de recursividad con el Registro de Disposiciones Jurídicas en tanto su función de certificación de vigencia. (Amoroso, 2015).

1.2.3 Gaceta Oficial

La Gaceta Oficial conocida también diario oficial, boletín oficial, registro oficial o periódico oficial, constituye el medio de comunicación escrito que un estado, organización internacional o regional, utiliza para publicar sus normas jurídicas, tales como: constituciones, tratados, leyes, decretos, reglamentos y otros actos de naturaleza pública. Una vez que las normas o decisiones han sido aprobadas o sancionadas por el o los respectivos órganos de poderes del Estado (poder legislativo, poder ejecutivo y poder judicial), deben ser promulgadas y publicadas para que tengan efectos jurídicos y por tanto sean acatadas y reconocidas como parte del ordenamiento jurídico del Estado. Esto con base en el principio de que la ley debe ser conocida por el público para que sea legítima. En algunas situaciones, particularmente en América Latina, cuando la norma aprobada requiere reserva por referirse a temas de la defensa nacional o labores de inteligencia, se publica una separata¹ de circulación restringida o reservada. En esos casos se habla de leyes secreta. (Amoroso, 2000.)

¹ **Separata:** Impresión por separado de un artículo o capítulo publicado en una revista o libro.

1.2.4 Publicidad electrónica

La publicación electrónica es un término que se utiliza con frecuencia, siendo este tipo de publicación usada cotidianamente. El concepto de publicación electrónica es entendido de diversas formas. Según Ángela Hechevarría, existen los siguientes enfoques (Hechavarría, 1997):

- Es la forma digitalizada de la publicación impresa.
- Es aquella que en su proceso de producción y edición incluye el uso de computadoras.
- Es la oficina sin papeles. Todos los documentos se encuentran en formato electrónico.
- Es toda publicación soportada en medios magnéticos.
- Es la que se disemina a través de redes.
- Es toda publicación que se caracteriza por su capacidad multimedia, por ser interactiva y por almacenarse y distribuirse en forma electrónica.

La publicación electrónica es una forma superior de publicación soportada en medios electrónicos y se apoya en las posibilidades que brindan los programas modernos para el tratamiento y acceso a la información. Estas ofrecen varias facilidades de interacción hombre/publicación de manera local o mediante el uso de redes de comunicación. (Hechavarría, 1997)

1.3 Tendencias actuales

Actualmente muchos países cuenta con soluciones para la edición y publicación electrónica de documentos legislativos. A continuación se exponen los resultados de un estudio sobre sistemas encargados de la publicación de normas jurídicas a nivel internacional que sirven de referencias para el desarrollo de la aplicación web que propone la autora del presente trabajo.

Gaceta Oficial del Estado Plurinacional de Bolivia: destinada a registrar las leyes, decretos y resoluciones supremas que promulgue el Poder Ejecutivo de la Nación. Una de las posibilidades que brinda es la de registrarse obteniendo así algunos beneficios como reporte de errores, sugerencias y tener un compendio de normas personalizado que le permita una mejor consulta de las mismas (gacetaoficialdebolivia.gob.bo, 2014) .

Gaceta Oficial de la República Bolivariana de Venezuela: tiene como función la publicación de leyes, reglamentos, acuerdos, circulares, órdenes y demás actos expedidos por el estado venezolano. Permite al usuario la consulta de los números ordinarios y extraordinarios de la Gaceta Oficial (gacetaoficialdelarepublicabolivarianadevenezuela.com, 2014).

Registro Oficial de Ecuador: se encarga de la difusión, publicación y control de las disposiciones legales que regirán a los ciudadanos que habitan en el territorio ecuatoriano, da a conocer las leyes para su vigencia y validez en el Estado de Ecuador (registroficial.gob.ec, 2014).

Boletín Oficial de la República de Argentina: además de la publicación de las leyes, permite el acceso a otros boletines de países de América Latina (boletinoficial.gov.ar, 2014).

Gaceta Oficial de la República de Cuba: El sitio de la Gaceta Oficial tiene como objetivo la publicación de disposiciones normativas, instrumentar nuevos recursos de información y diversificar las opciones de los destinatarios de la información publicada por la GO, para acceder a ella. Además se encarga de conservar en soporte digital el tracto histórico de las normas jurídicas publicadas por la Gaceta, con vistas a hacerlo más eficiente, efectivo, y potenciar las prestaciones de su archivo central (Ivonet, y otros, 2014).

Señalar que estas soluciones no permiten dar respuesta al problema de inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS, debido a que las mismas solo realizan la publicación de actos normativos. Siendo necesario el desarrollo de una aplicación web que mitigue los problemas antes mencionados, teniendo en cuenta las características de las soluciones estudiadas

Como resultado del estudio antes realizado, se identificaron un conjunto de características y funcionalidades que pueden ser incorporadas en la solución propuesta, destacándose la forma en que se publican las leyes, resoluciones, circulares y demás actos normativos promulgados en estos países. Además permitió obtener experiencia en cuanto a la manera de estructurar y organizar los contenidos en aplicaciones web de este tipo.

1.4 Metodología de desarrollo de software

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores en la realización del software (Menéndez-Barzanallana, 2006).

Existen dos grandes grupos de metodologías, las tradicionales o robustas y las ágiles. La primera está asociada a un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada que se genera en cada una de las actividades desarrolladas. La misma es más efectiva y necesaria en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige mayor formalidad en el desarrollo del software (Penadés, y otros, 2003).

CAPÍTULO 1

Por otro lado las metodologías ágiles están enfocadas a proyectos donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo. Las metodologías ágiles están orientadas a proyectos pequeños y constituyen una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto (Penadés, y otros, 2003).

A partir de lo analizado y teniendo en cuenta que el equipo de trabajo es pequeño (de un integrante). Además que la solución propuesta puede estar sujeta a frecuentes cambios durante su desarrollo y existe un constante intercambio con el cliente, siendo este parte del equipo, se decide emplear una metodología ágil para el desarrollo de la aplicación web.

Entre las metodologías ágiles más usadas se encuentran Extreme Programming (XP), Scrum, Crystal Clear, Adaptive Software Development (ASD), Feature Driven Development (FDD), Lean Development y Dynamic Systems Development Method (DSDM). En la Tabla 1 se muestra una comparación de estas metodologías, donde los resultados y valores más altos representan una mayor agilidad, en base a los parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo, simplicidad, excelencia técnica, adaptabilidad y prácticas de colaboración.

Tabla 1: Metodologías ágiles más usadas (Highsmith, 2002).

	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	5	4	3	3	4	5	5
Colaboración	5	5	4	4	4	5	5
Adaptabilidad	5	5	3	3	4	4	3
Simplicidad	4	4	3	5	3	5	5
Excelencia técnicas	3	3	4	4	4	3	4
Prácticas de colaboración	5	5	4	3	3	4	5

Teniendo en cuenta los valores de la tabla, las características del desarrollo de la solución propuesta y que los especialistas de la Universidad de la Ciencias Informáticas (UCI) tienen gran dominio sobre la utilización de XP, existiendo documentación necesaria para poder realizar un buen uso de la misma, se decide utilizar esta metodología en el desarrollo del presente trabajo.

1.4.1 Extreme Programming (XP)

La Programación Extrema o Extreme Programming (XP), según Andujar, es la más destacada de las metodologías ágiles. La metodología XP está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. También se define como adecuada para proyectos con requisitos imprecisos y donde existe un alto riesgo técnico (Andujar, y otros, 2013).

Según Priolo (Priolo, 2007), la metodología de desarrollo XP promueve un conjunto de valores que permite al equipo de desarrollo enfrentar cambios imprevistos durante el proceso de desarrollo y obtener un producto con exitoso.

El ciclo de vida de XP cuenta de seis fases: (Priolo, 2007)

Exploración: Los clientes plantean a grandes rasgos las historias de usuario y el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. En la fase se genera como artefacto las Historias de Usuario (HU).

Planificación de la entrega: el cliente establece la prioridad de cada historia de usuario, y correspondientemente los programadores realizan una estimación del esfuerzo necesario por cada una de ellas. Además se determina un cronograma en conjunto con el cliente. El artefacto generado durante la fase es la Estimación de esfuerzo por HU.

Iteraciones: esta fase se realiza varias iteraciones sobre el sistema antes de ser entregado. El cliente escoge las historias que se implementarán en cada iteración, todo el trabajo de las iteraciones es expresado en tareas. Durante la fase se realiza también el análisis y diseño de la aplicación. Los artefactos generados en la misma son: el Plan de duración por iteración, las Tarjetas CRC (Clase, Responsabilidad y Colaborador) y las Tareas de ingeniería.

Producción: se realizan pruebas y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente, determinándose errores, modificaciones en el mismo. Los artefactos generados durante la fase son: el Acta de liberación del producto y el Acta de aceptación.

Mantenimiento: el sistema debe mantenerse en funcionamiento al mismo tiempo que desarrolle nuevas iteraciones. Esta fase puede requerir de nuevo personal o cambios en la estructura del equipo.

Muerte del Proyecto: el usuario no tiene más historias de usuario para incluir en el sistema, y se han satisfecho las necesidades del cliente en otros aspectos como rendimiento y confiabilidad.

Cada una de las fases antes descritas es tomada en cuenta para guiar el desarrollo de la solución propuesta.

1.5 Herramienta CASE.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) constituyen un conjunto de herramientas y métodos informáticos destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y costos. Estas herramientas han sido diseñadas para proporcionar asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Suelen incluir un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos, herramientas para el diseño y el modelo de datos corporativo, así como los esquemas conceptuales y lógicos. También posibilitan el diseño de prototipos. Existen varias herramientas CASE entre las que se encuentran, Erwin, EasyCASE, Oracle Designer, System Architect y Visual Paradigm (Meza, 2011).

Para el desarrollo del presente trabajo se selecciona Visual Paradigm en su versión 8.0 para la modelación, ya que es una herramienta multiplataforma, fácil de usar, portable y posee una interfaz agradable que permite una mejor interacción. Además la UCI cuenta con una licencia para su uso.

Visual Paradigm

Visual Paradigm es una herramienta CASE que ofrece una gama de funcionalidades que permiten el modelado de artefactos generados durante el ciclo de vida del desarrollo de software. Permite construir diagramas de diversos tipos y provee soporte para la generación de código inverso, generar código desde diagramas y generar documentación. Además permite la integración con otros IDE como: Eclipse, NetBeans, JBuilder y JDeveloper. También brinda facilidades para integrarse con la notación BPMN (Business Process Modeling Notation, en español Notación para el Modelado de Procesos de Negocios) (Trujillo, y otros, 2013).

1.6 Lenguaje de Modelado Unificado (UML)

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language), tal como indica su nombre es un lenguaje de modelado que tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases y objetos, hasta la implementación y configuración con los diagramas de despliegue. Es independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML. (Orallo, 2012).

Ventajas del uso de UML (Orallo, 2012):

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos, permite generar código fuente a partir de los modelos o realizar una ingeniería inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados.

La inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS, son procesos sencillos y fáciles de comprender. Teniendo en cuenta las características y ventajas mencionadas sobre el lenguaje de modelado UML, la autora del presente trabajo decide utilizar el mismo en la modelación y generación de los artefactos obtenidos en el desarrollo de la presente investigación.

1.7 Sistemas Gestor de contenido (CMS)

Un Sistema Gestor de Contenidos (CMS por siglas en inglés Content Management System), también conocidos como Gestores de Contenido Web (WCM), es una herramienta que permite llevar a cabo los procesos de creación, almacenamiento, tratamiento y presentación de la información mediante una aplicación web. Además posibilita la gestión de los contenidos de manera uniforme, accesible, y cómoda para el usuario. Se caracterizan por posibilitar la inmediatez de la publicación y edición de los contenidos, constituyendo una ventaja con respecto a su uso. (Sarduy, y otros, 2006).

Ventajas de los CMS:

- La capacidad de manejar eficientemente gran cantidad de páginas web.

- Trabajar en un ambiente de páginas web interactivas, es decir, que se generan según las peticiones de los usuarios.
- Controlar el acceso de los usuarios al sistema, no sólo mediante su contraseña, sino mediante los permisos asignados a cada uno y la información que incluye, tanto en calidad como en cantidad, que posibilita el crecimiento y desempeño del sistema.
- Orden en el sistema, al existir la posibilidad de asignar, por parte de la herramienta, un mismo estilo a todas las páginas generadas (Sarduy, y otros, 2006).

A continuación se presenta un análisis de los CMS más usados y de mayor relevancia en la actualidad (WordPress, Joomla y Drupal), debido a que son de código abierto y cuentan con una extensa documentación para su estudio y empleo.

1.7.1 WordPress.

WordPress es una avanzada plataforma semántica de publicación personal orientada a la estética, los estándares web y la usabilidad. Desarrollado en Hypertext Preprocessor (PHP) y My Structured Query Language (MySQL), y licenciado bajo GNU/GPL v2 (o posterior). Es un proyecto de código abierto, lo que significa que los usuarios son libres de usarlo para cualquier cosa de su sitio sin tener que pagar derecho de licencia. Se ha convertido en uno de los CMS más populares de la Blogosfera, ya que pone especial atención a la elegancia, la estética, los estándares web y la usabilidad. Dentro de sus principales características se encuentran: su fácil instalación, actualización, personalización y actualización automática del sistema. Una de las causas de su enorme crecimiento es debido a que cuenta con una gran comunidad de desarrolladores y diseñadores, que se encargan de desarrollarlo en general o crear plugins y temas para la comunidad (wordpress.org, 2015).

1.7.2 Joomla

Es uno de los CMS más usados de código abierto y está disponible bajo licencia GNU/GPL. El código base de Joomla está escrito en lenguaje script PHP y utiliza como base de datos MySQL. Dentro de sus principales rasgos esta su facilidad de instalación, compresión y caché para mejorar el rendimiento (Joomla, 2005).

Funciones Generales de Joomla (JoomlaSpanish, 2005):

- Gestión del contenido del sitio basado completamente en una base de datos.
- Todas las secciones de noticias, productos o servicios se pueden editar y gestionar.
- Las secciones de temas pueden ampliarse mediante aportaciones de autores.
- Las características de etiquetado de los contenidos permiten un acceso flexible para cada tipo de usuario.

- Diseños completamente personalizables, incluyendo los menús izquierdo, derecho y central.

1.7.3 Drupal

Es uno de los CMS más utilizados por la calidad del código, su estabilidad, su flexibilidad y adaptabilidad. Es una herramienta de gestión de contenido dinámica, de código abierto construido sobre PHP y distribuido bajo los términos de la licencia GNU/GPL. Cuenta con una activa comunidad de usuarios en función de su mantenimiento. Permite publicar artículos, imágenes, archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Características generales de Drupal (Drupal, 2005):

- Ayuda on-line: Un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- Búsqueda: Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- Código abierto: El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de blogs o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades de los desarrolladores.
- Módulos: La comunidad de Drupal ha desarrollado muchos módulos que proporcionan funcionalidades como autenticación mediante jabber y mensajes privados.
- Personalización: Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.
- Autenticación de usuarios: Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio.
- Permisos basados en roles: Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles.
- Independencia de la base de datos: Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

1.7.4 Selección del sistema de gestión de contenido a utilizar

Después del análisis de las principales características de los CMS antes descritos la autora del presente trabajo decide utilizar el CMS Drupal en su versión 7.34 en el desarrollo de la aplicación informática que da respuesta al objetivo general de la presente investigación. Los elementos tenidos en cuenta fueron: que ofrece las prestaciones y funcionalidades necesarias para gestionar los contenidos de un portal web, proporciona gestión de usuarios, clasificación y edición de contenidos y posee un sistema de seguridad basado en roles, que permite controlar los diferentes módulos según los permisos que el administrador definida para cada rol. También permite la combinación de módulos, que pueden ser modificados según las necesidades de los desarrolladores que los utilizan.

1.8 Lenguajes de programación

Un lenguaje de programación constituye un sistema estructurado y diseñado principalmente para que las computadoras se entiendan entre sí y con los humanos. Contiene un conjunto de acciones consecutivas que el ordenador debe ejecutar. En la actualidad hay más de 100 lenguajes de programación diferentes. (areatecnologia.com, 2015) A continuación se describen los lenguajes de programación que se utilizarán en el desarrollo de la aplicación.

1.8.2 Lenguaje programación del lado del servidor

Un lenguaje de programación del lado del servidor es aquel que es reconocido, ejecutado e interpretado por el servidor y que se envían al cliente en un formato comprensible para él. Estos lenguajes se utilizan para generar páginas HTML (Hyper Text Markup Language) dinámicas, facilitando el acceso a las bases de datos.

Hypertext Preprocessor (PHP)

PHP ²es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web. La meta principal de este lenguaje es generar páginas web rápidas y dinámicas. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que puede recopilar datos de formularios, generar páginas con contenidos dinámicos o enviar y recibir cookies³. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF⁴ e incluso películas Flash generadas sobre la marcha. También

²PHP: (acrónimo recursivo de PHP: Hypertext Preprocessor)

³ Cookies: es un apunte que un sitio web guarda en el navegador y solo son leídas por el sitio web que las creó. Sirven para que un sitio web recuerde quién eres y qué quieres.

⁴ PHP: Portable Document Format.

se puede generar fácilmente cualquier tipo de texto, como XHTML⁵ y cualquier otro tipo de fichero XML⁶. Puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (Solaris, OpenBSD, otros), Windows, Mac OS X, y también admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Una de las características más destacables de PHP es su soporte para un amplio abanico de bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras (Olson, 2015).

1.8.3 Lenguajes de programación del lado del cliente

En la programación del lado del cliente el navegador es el encargado de interpretar y mostrar por pantalla la página creadas, el servidor no interviene en el proceso de crear la página web solicitada por el usuario.

HTML

HTML es un lenguaje de marcado que permite indicar la estructurar y presentar el contenido para la web mediante etiquetas. Es independiente de cualquier plataforma y permite combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas. Este lenguaje ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas, ya que siguen un esquema de árbol donde existe un elemento raíz y donde el resto de etiquetas se insertan de manera lógica y estructurada. Además, se trata de un fichero de texto, y solo se necesita un editor como el bloc de notas para generar un documento HTML, permite la combinación dentro de su código de otros lenguajes como PHP y JavaScript (Ruiz, 2014).

Cascading Style Sheets (CSS)

Hojas de Estilo en Cascada o CSS (por sus siglas en ingles Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Esta forma de descripción de estilos permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (w3c.es, 2014).

⁵ **HTML**: eXtensible Hypertext Markup Language.

⁶ **XML**: eXtensible Markup Language

JavaScript

JavaScript (a veces abreviado como JS) ha sido desarrollado por Netscape, es un lenguaje ligero e interpretado, orientado a objetos más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Permite agregar funcionalidades, validar formularios, comunicarse con el servidor. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. En ocasiones muchos suelen confundir Java Script con lenguaje de programación Java, pero son lenguajes totalmente diferentes. JavaScript es interpretado por lo que no requiere de compilación. (developer.mozilla, 2014).

1.9 Sistemas Gestores de Base de Datos

Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Debe permitir:

- Definir una base de datos: especificar tipos, estructura, y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio que pueda ser controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Algunas características deseables en un Sistema Gestor de Base de Datos son:

- Control de redundancia: la redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdiciar espacio en disco, puede provocar inconsistencia de datos), aunque a veces es deseable por cuestiones de redundancia y seguridad.
- Restricción de los accesos no autorizados: cada usuario debe tener permisos de acceso.
- Cumplimiento de las restricciones de integridad: SGBD debe ofrecer recursos para definir y garantizar el cumplimiento de la integridad. (Slideshare.net, 2012)

1.9.2 PostgreSQL

PostgreSQL es un sistema de base de datos, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección. Tiene soporte completo para claves foráneas, combinaciones, vistas, triggers y procedimientos almacenados (en varios

idiomas). También es compatible con el almacenamiento de grandes objetos binarios, como imágenes, sonidos, o de vídeo (Postgresql.org, 2010).

Sus características técnicas lo hacen uno de los SGBD más potentes y robustos del mercado. A continuación se ofrece algunas de las características de PostgreSQL (Postgresql.org, 2010):

- El código fuente está disponible para todos sin costo, haciéndolo extensible.
- Es una base de datos 100% ACID⁷.
- Licencia BSD⁸.
- Copias de seguridad en caliente (Online/hot backups)
- Acceso encriptado vía SSL⁹.
- Integridad referencial, para garantizar la validez de los datos de la base de datos
- Multi-Version Concurrency Control (MVCC): una estrategia de almacenamiento de filas para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit.

1.9.3 MySQL

MySQL es un sistema gestor de base de datos relacional, caracterizado por su velocidad y robustez. Las prestaciones que caracterizan a este gestor de base de datos se mencionan a continuación (Casillas, y otros, 2013):

- Es multiplataforma.
- Está optimizado para equipos de múltiples procesadores, es multihilo.
- Se puede utilizar como cliente-servidor.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.

⁷ **ACID**: es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

⁸ **Licencia BSD**: Es una licencia de software libre permisiva otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Una aplicación licenciada con BSD permite que otras versiones puedan tener otros tipos de licencias, tanto propietarias como libres.

⁹ **SSL**: (Secure Sockets Layer o Capa de Enchufe Seguro): es un protocolo criptográfico diseñado en 1994 por la empresa Netscape Communications Corporation, que proporciona comunicaciones seguras por una red, comúnmente Internet.

- Los mensajes de error están en español y pueden hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.

Posee licencia dual, por un lado utiliza la licencia GNU/GPL y por otro se distribuye una versión comercial. Esta última se diferencia de la versión libre en el soporte técnico y la posibilidad de integrar este gestor en un producto privativo, siendo esto una desventaja ya que el copyright del código es propiedad del autor individual y no ofrece soporte para la versión libre.

1.9.4 SQLite

SQLite es una biblioteca de software que implementa SGDB transaccionales SQL auto-contenido, sin servidor y sin configuración. SQLite es el motor de base de datos SQL de mayor despliegue en el mundo y su código fuente es de dominio público (SQLITE, 2015).

Características principales de SQLite (SQLite, 2015):

- SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas, contenida en un único archivo.
- Las transacciones son atómicas, consistente, aislado, y durable incluso después bloqueos del sistema y fallas de energía.
- No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro.
- Cero-configuración, no necesita de configuración ni administración.
- La base de datos se almacena en un único archivo multiplataforma.
- Huella de código pequeño: menos que 500KiB totalmente configurado o mucho menos si las características opcionales se omite.
- API¹⁰ simple, fácil de usar y de compilar, y por lo tanto es fácil de añadir a un proyecto más amplio.
- Autónomos: no hay dependencias externas.
- El código es de dominio público. Uso para cualquier propósito.

1.9.5 Selección del sistema de gestión de base de datos a utilizar

Después de analizados los SGBD: PostgreSQL, SQLite y MySQL se decide utilizar en el desarrollo de la solución propuesta MySQL en su versión 5.6.12, teniendo en cuenta su

¹⁰ **API:** siglas en inglés de Application Programming Interface (Interfaz de Programación de Aplicaciones).

velocidad, fiabilidad, robustez, facilidad de instalación y configuración. Además de soportar gran cantidad de tipos de datos para las columnas y ser compatible con el CMS Drupal. Estando en correspondencia con las características de la solución a desarrollar.

1.10 Entorno de desarrollo integrado.

Un entorno de desarrollo integrado (Integrated Development Environment: IDE) es un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un editor de código, administrador de proyectos, archivos, enlace transparente a compiladores e integración con sistemas controladores de versiones o repositorios. Estas series de herramientas las utilizan los programadores para desarrollar código y están pensadas para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos (Pedraza, y otros, 2014).

A continuación se describe el IDE definido para el desarrollo de la presente investigación.

1.10.1 Netbeans IDE

NetBeans IDE es un entorno de desarrollo para que programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en java pero proporciona un conjunto de herramientas y editores para otros lenguajes como XML, HTML, PHP, C / C ++, JavaScript y JSP. Permite rápida y fácilmente desarrollar un escritorio Java, aplicaciones web y para móvil, interfaz gráfica de forma visual, además de proporcionar soporte para el control de versiones. Se puede instalar en todos los sistemas operativos compatibles con Java, de Windows a Linux para sistemas Mac OS X. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo (Netbeans.org, 2015).

Teniendo en cuenta lo antes expuesto y el frecuente uso de este IDE en los proyectos productivos de la UCI, la autora del presente trabajo decide utilizar Netbeans en su versión 7.3 para el desarrollo de la solución propuesta.

1.11 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Un servidor web está diseñado para transferir hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

A continuación se describe el servidor web definido para el desarrollo de la presente investigación.

1.11.1 Apache Server

Apache Server es uno de los más robustos y rápidos servidores web multiplataforma que existen. Este servidor HTTP (Hypertext Transfer Protocol) creado en 1995 bajo la idea de open-source está disponible para sistemas operativos modernos, incluyendo Unix y Windows NT. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado (del Castillo, 2000).

Entre sus características se destacan (del Castillo, 2000):

- Multiplataforma
- Es un servidor web conforme al protocolo HTTP.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

Teniendo en cuenta lo antes expuesto la autora del presente trabajo decide utilizar el servidor web Apache 2.4 para el desarrollo de la solución propuesta.

1.12 Conclusiones parciales

- El análisis de conceptos relacionados con los procesos de inscripción y revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS, facilitó una mejor comprensión del marco teórico de la investigación.
- El estudio de soluciones existentes en la actualidad para la publicación de disposiciones jurídicas en el área de Latinoamérica, demostró que los trabajos realizados, no dan respuesta a la problemática que da origen a la presente investigación, dando lugar al desarrollo de la aplicación web propuesta.
- La selección de la metodología XP y las distintas herramientas, permiten obtener una aplicación web acorde con las necesidades del cliente.

Capítulo 2: Análisis y diseño de la aplicación web.

2.1 Introducción

En el siguiente capítulo se describen los principales elementos tenidos en cuenta en el desarrollo de la solución propuesta, a partir del patrón arquitectónico utilizado y los patrones de diseño empleados. Se hace referencia también a los principales artefactos generados con el uso de la metodología seleccionada, entre las que se destacan: Historias de usuarios (HU), Estimación de esfuerzo por HU y Plan de iteraciones. Otros de los elementos analizados en el capítulo son los roles definidos a los usuarios para el acceso a la aplicación web.

2.2 Descripción de la aplicación web

La propuesta de solución de la presente investigación, constituye una aplicación web que permite realizar la inscripción, revisión, y control de vigencia de las disposiciones jurídicas inscriptas en Registro de Disposiciones jurídicas del MINJUS y publicadas en la Gaceta Oficial de la República de Cuba. La aplicación se realiza incorporando las facilidades que ofrece el desarrollo web actual, en la mejora del canal de comunicación y servicios entre los usuarios que intervienen en la emisión y recepción de disposiciones en el Registro General de Disposiciones Jurídicas del MINJUS y el sitio de la Gaceta Oficial de la República de Cuba.

La aplicación web permite comprobar si cada una de las disposiciones jurídicas publicadas en la GO se encuentra vigente, a partir de la fecha de publicación y el tiempo en que deben entrar en vigor las mismas. Además ofrece un trabajo colaborativo entre las instituciones que intervienen en el proceso legislativo cubano, facilita el proceso de edición de disposiciones jurídicas para ser publicadas en la GO, permite introducir mejoras en la certificación de leyes, resoluciones, entre otras normas inscriptas en el Registro General de Disposiciones Jurídicas del MINJUS. En la Figura 1: **Vista general de**Vista general de la aplicación web, se representa cómo quedan organizados cada uno de los contenidos a mostrar en la solución obtenida.

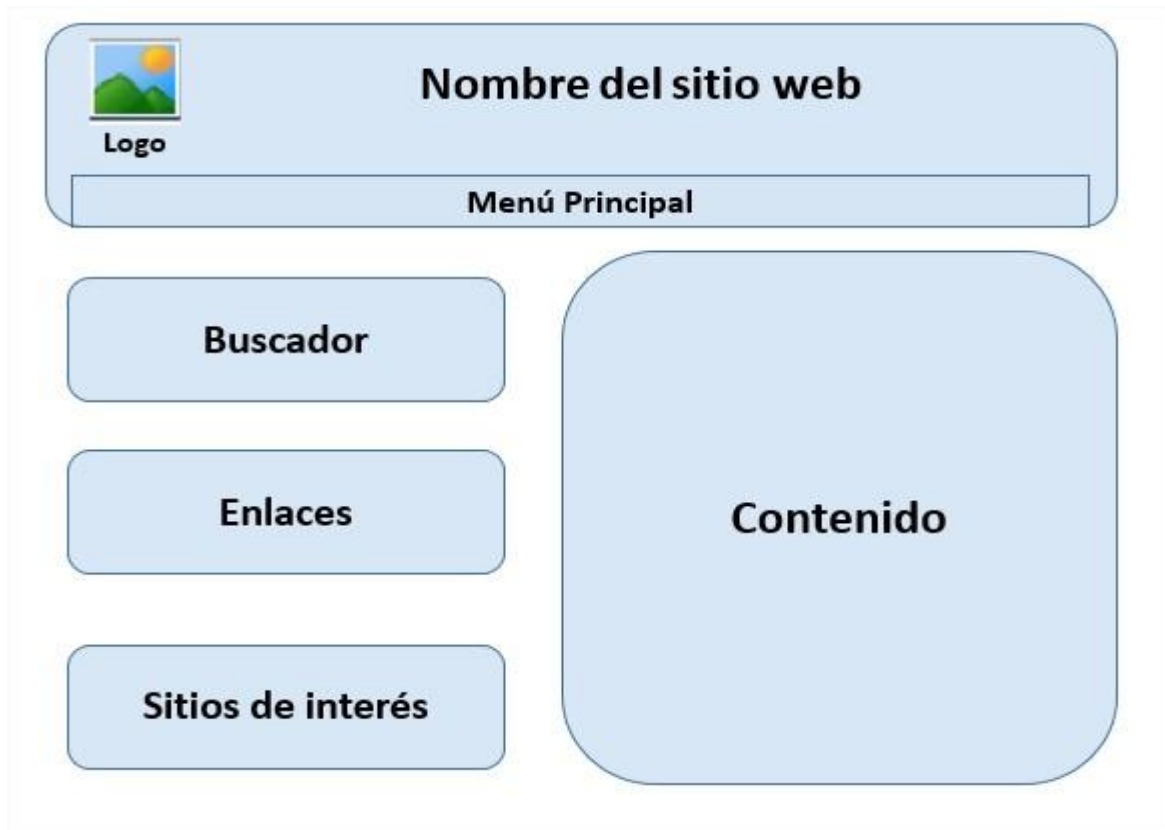


Figura 1: Vista general de la aplicación web.

La solución cuenta además con cinco niveles de usuarios, los cuales se describen a continuación:

Administrador: posee control total del sistema. Será el encargado de gestionar los permisos de los usuarios y la administración de la aplicación en general.

Revisor: Persona que presenta los privilegios necesarios para revisar los documentos enviados por los emisores oficiales de disposiciones jurídicas y velar porque estos cumplan con los requisitos formales.

Promulgador: Persona que presenta los privilegios necesarios para registrar una ley, puede ser de la Fiscalía General de la Republica (FGR), del Tribunal Supremo Provincial (TSP) o de los Organismos de la Administración Central del Estado (OACE).

Editor: Persona que presenta los privilegios necesarios para realizar cambios en los documentos legislativos a publicar en la GO.

Publicador: Persona que tendrá el privilegio de publicar cualquier documento, así como, consultar cualquier otro artículo.

2.3 Fases del proceso de desarrollo

El ciclo de vida de XP consta de 6 fases. A continuación se describe cómo quedó organizado el desarrollo del presente trabajo, en las fases de Exploración, Planificación de la entrega e Iteraciones.

2.3.1 Exploración

En esta fase se definen las HU, en las cuales se describen los requisitos funcionales de interés para el cliente. En la fase también se describen los requisitos no funcionales de la aplicación. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas a utilizar en el proyecto.

2.3.1.1 Requisitos del sistema

Definen el comportamiento que debe tener la aplicación web, minimizando la posibilidad de errores y especificando de forma clara lo que el cliente desea. Mostrando así las cualidades, características, funciones y atributos que debe cumplir la solución propuesta. Se dividen en requisitos funcionales (RF) y requisitos no funcionales (RNF).

Requisitos funcionales

Los requisitos funcionales son aquellos que indican lo que debe hacer la solución propuesta, son las capacidades con las que debe cumplir la misma. A continuación se listan, organizados por HU, los RF identificados para el desarrollo de la aplicación web propuesta:

HU: Autenticar usuario.

RF1: Autenticar usuario.

HU: Gestionar usuario.

RF2: Modificar usuario.

RF3: Insertar usuario.

RF4: Eliminar usuario.

HU: Gestionar roles.

RF5: Modificar rol.

RF6: Adicionar rol.

RF7: Eliminar rol.

HU: Verificar fortaleza de contraseña de usuario.

RF8: Verificar fortaleza de contraseña de usuario.

HU: Cerrar sesión.

RF9: Cerrar sesión.

HU: Gestionar disposición jurídica en el Registro General de Disposiciones Jurídicas.

RF10: Registrar disposición jurídica.

RF11: Modificar disposición jurídica.

RF12: Eliminar disposición jurídica.

HU: Mostrar el listado de las disposiciones jurídicas publicadas en el registro

RF13: Mostrar el listado de las disposiciones jurídicas publicadas en el registro.

HU: Control de las vigencias de las disposiciones jurídicas.

RF 14: Control de las vigencias de las disposiciones jurídicas.

HU: Mostrar información de la Gaceta Oficial.

RF16: Mostrar información de la Gaceta Oficial.

HU: Gestionar Comentario.

RF17: Enviar comentarios.

RF18: Aprobar comentarios.

RF19: Mostrar comentarios.

Requisitos no funcionales:

Los RNF son restricciones de los servicios o funciones ofrecidas por el sistema, incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares. Los RNF se refieren a las propiedades emergentes que debe cumplir el software, tales como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Apariencia o Interfaz externa:

RNF 1: El sistema debe presentar una interfaz de fácil interacción, permitiendo que el usuario llegue de manera rápida y efectiva a la información buscada. Contará con menús para agilizar y acelerar su utilización.

RNF 2: La interfaz de comunicación de la aplicación debe ser segura y confiable.

RNF 3: El sistema estará optimizado para una resolución de 1024x768 píxeles.

Usabilidad:

RNF 4: Se proporcionará una interfaz sencilla que posibilite una rápida adaptación, tanto para los usuarios con conocimientos avanzados en informática como los usuarios más inexpertos.

RNF 5: Debe tener buena visibilidad en los siguientes navegadores web Firefox versión 35 o superior, Internet Explorer versión 5 o superior y Chrome.

Eficiencia:

RNF 6: El sistema debe ser capaz de responder con rapidez a las peticiones de los usuarios.

Seguridad:

RNF 7: El acceso será controlado con nombres de usuario y contraseñas.

RNF 8: La información manejada por el sistema estará protegida del acceso y divulgación no autorizada.

RNF 9: Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.

RNF 10: El usuario debe ser identificado antes de realizar cualquier acción sobre el contenido del sistema. Garantizar que cada usuario realice las actividades permitidas a su rol.

RNF 11: El acceso a la información del sistema será de acuerdo al nivel de roles establecido.

Soporte:

RNF 12: Servidor web Apache 2.2.21 o superior.

RNF 13: Gestor de base de datos MySQL en su versión 5.4.12 o superior.

RNF 14: Debe dar la posibilidad de ser mejorado y de incorporarle nuevos servicios en caso de necesitarlos.

Software:

RNF 15: Por parte del cliente se requiere un navegador web capaz de interpretar HTML 5, JavaScript 1.6 y CSS 3.

RNF 16: El uso de la aplicación es multiplataforma.

Hardware:

RNF 17: Los servidores proxy, web y de base de datos deben poseer como mínimo 512 MB de memoria RAM.

2.3.1.2 Historias de usuarios

Las HU es la técnica utilizada en XP para indicar las funciones que debe realizar el sistema. Típicamente deben ser escritas en tarjetas donde se incluye una breve descripción de las funcionalidades que debe cumplir el sistema. Las HU se crean y modifican a medida que el proyecto avanza, permiten responder rápidamente a los requisitos cambiantes. A continuación se listan las HU que agrupan cada uno de los RF definidos para el desarrollo de la aplicación web.

1. Autenticar usuario.
2. Gestionar usuario.
3. Gestionar roles.
4. Verificar fortaleza de contraseña de usuario.
5. Cerrar sesión.
6. Gestionar disposición jurídica en el Registro General de Disposiciones Jurídicas.
7. Control de vigencias de las disposiciones jurídicas.
8. Mostrar el listado de las disposiciones jurídicas publicadas en el registro
9. Mostrar información de la Gaceta Oficial.
10. Gestionar Comentario.

CAPÍTULO 2

En las tablas 2 y 3 se describen la HU “Gestionar disposición jurídica en el RGDJ” y la HU “Control de vigencias de las disposiciones”, las cuales fueron seleccionadas teniendo en cuenta su complejidad y que describen las principales funcionalidades de la aplicación web.

Tabla 2: Historia de Usuario: Gestionar disposición jurídica en el RGDJ.

Historia de Usuario	
Número: 6	Nombre de la Historia de Usuario: Gestionar disposición jurídica en el Registro General de Disposiciones Jurídicas.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Dailenis Alcántara Frómata.	Iteración asignada: 1.
Prioridad en negocio: Alta.	Puntos estimados: 3 semanas.
Riesgo en desarrollo: Alta.	Puntos reales: 3 semanas.
Programador responsable: Dailenis Alcántara Frómata.	
Descripción: Consiste en inscribir un disposición jurídica en la aplicación con los diferentes campos que tiene, así como modificar uno que no se haya terminado y tenga estado de editado”.	

Tabla 3: Historia de Usuario: Control de vigencias de las disposiciones.

Historia de Usuario	
Número: 7	Nombre de la Historia de Usuario: Control de vigencias de las disposiciones.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Dailenis Alcántara Frómata.	Iteración asignada: 1.
Prioridad en negocio: Alta.	Puntos estimados: 3 semanas.
Riesgo en desarrollo: Alta.	Puntos reales: 3 semanas.
Programador responsable: Dailenis Alcántara Frómata.	
Descripción: Consiste en determinar el periodo de vigencia de las disposiciones legales, si ha entrado en vigor o no, así como el tiempo de aplazamiento de esta, además del estado de la misma.	

2.3.2 Planificación de la entrega.

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días, identificándose el número y tamaño de las iteraciones.

2.3.2.1 Estimación de esfuerzo por historias de usuarios

Los puntos de estimación de esfuerzo por HU constituyen el tiempo de implementación estimado para cada una de estas, cada punto representa una semana ideal de trabajo. Las HU deben ser programadas en un tiempo estimado hasta de 3 semanas. Si la estimación supera las 3 semanas la HU deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. En caso de que el esfuerzo sea menor que una semana, la HU será combinada por otra. La siguiente tabla muestra la estimación de esfuerzos por HU para el desarrollo de la aplicación web propuesta.

Tabla 4: Estimación de esfuerzo por Historia de Usuario

Historia de Usuario	Puntos de Estimación
Autenticar usuario.	2
Gestionar usuario.	
Gestionar roles.	
Verificar fortaleza de contraseña de usuario.	1
Cerrar sesión.	
Gestionar disposición jurídica en el Registro General de Disposiciones Jurídicas.	3
Mostrar el listado de las disposiciones jurídicas publicadas en el registro.	1

Control de las vigencias de las disposiciones jurídicas.	3
Mostrar información de la Gaceta Oficial	1
Gestionar Comentario	1

Luego de realizada la estimación de esfuerzo por HU, se concluye la fase de Planificación de la entrega, dando paso a la fase Iteraciones.

2.3.3 Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado al cliente. El plan de entrega está compuesto por iteraciones con una duración de no más de tres semanas. En la primera iteración se puede establecer la arquitectura utilizada a lo largo del proyecto (como valor agregado al negocio). Al final de la última iteración la aplicación web estará listo para ser entregada al cliente.

2.3.3.1 Plan de duración de las iteraciones.

Después de tener la estimación de esfuerzo por historias de usuarios es necesario crear el plan de duración para cada una de las iteraciones definidas, que tiene como objetivo mostrar la duración y el orden en que serán implementadas las HU dentro de cada iteración. Para la implementación de la aplicación web propuesta se necesita un total de 12 semanas, divididas en 2 iteraciones, como se muestra en la tabla 6.

Tabla 5: Plan de duración de las iteraciones.

Iteración	Orden de la HU a implementar	Duración
Iteración # 1	HU: Autenticar usuario. HU: Gestionar usuario. HU: Gestionar roles. HU: Gestionar disposición jurídica en el Registro General de Disposiciones Jurídicas.	9 semanas

	<p>HU: Mostrar el listado de las disposiciones jurídicas publicadas en el registro</p> <p>HU: Control de las vigencias de las disposiciones jurídicas.</p>	
Iteración # 2	<p>HU: Mostrar información de la Gaceta Oficial.</p> <p>HU: Gestionar Comentario.</p> <p>HU: Verificar fortaleza de contraseña de usuario.</p> <p>HU: Cerrar sesión.</p>	3 semanas

2.3.4 Arquitectura del sistema

La arquitectura empleada en el desarrollo de la aplicación web se define a partir de la utilizada por el CMS Drupal. Esta es una arquitectura modular que permite ampliar sus funcionalidades a través de métodos uniformes de desarrollo e integración de nuevos módulos. La misma está basada en el patrón arquitectónico Modelo-Vista-Controlador (M-V-C), el cual divide la aplicación en tres partes: el modelo, la vista y el controlador, separando la lógica de negocio, de la interfaz de usuario y los datos de la aplicación. La Figura 2 muestra a través del diagrama de componentes de la solución propuesta, el uso del patrón arquitectónico M-V-C.

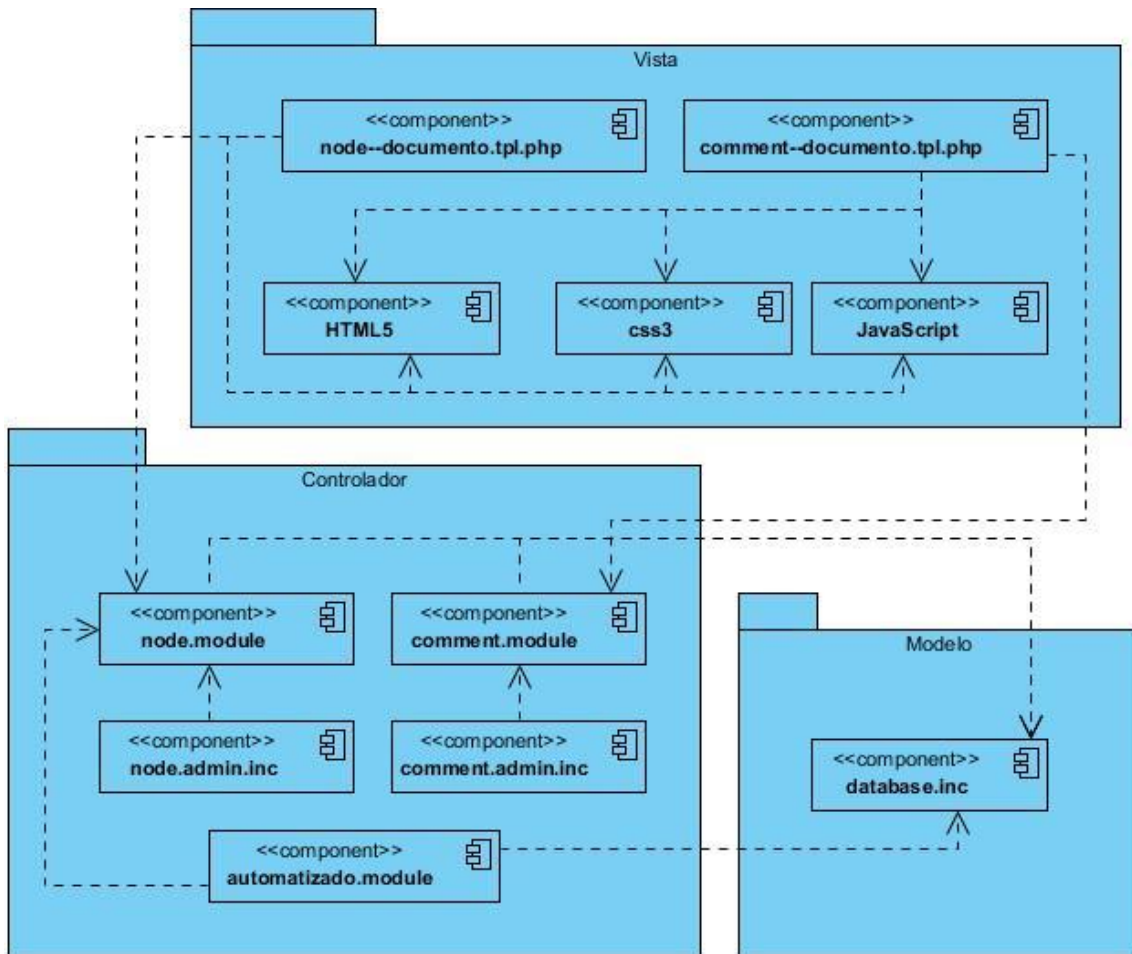


Figura 2: Diagrama de componentes.

A continuación se describen los elementos representados en la figura 2 sobre el empleo del patrón arquitectónico M-V-C en el desarrollo de la solución propuesta.

Modelo: Agrupa las entidades que representan el modelo de datos en el cual se almacena toda la información manejada por la aplicación web (documentos legislativos, comentarios y artículos)

Vista: Es la responsables de agrupar los elementos de diseño relacionados con la interfaz de usuario de la aplicación web, tales como páginas HTML, estilos CSS, validaciones Js e imágenes, los cuales se encargan de mostrar la información al usuario transformando el modelo en una página web.

Controlador: se ocupa de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista (archivos de Drupal con extensión .module), sirviendo de intermediaria entre la vista y el modelo.

2.4 Diseño de la aplicación web

La metodología de desarrollo de software XP propone que se realicen diseños simples y sencillos, que permita un fácil entendimiento a los desarrolladores, permitiendo reducir el tiempo al realizar las tareas de implementación. Esta metodología estimula el empleo de tarjetas CRC (Clase-Responsabilidad-Colaborador) para guiar el diseño en el proceso de desarrollo.

2.4.1 Tarjetas CRC (Clase, Responsabilidad y Colaboración)

Las tarjetas CRC son una técnica para el diseño de software orientado a objetos creada por Kent Beck y Ward Cunningham. Las tarjetas CRC se utilizan, individualmente, para representar objetos. La clase del objeto puede ser escrita en la parte superior de la tarjeta, las responsabilidades se colocan en la parte izquierda y las clases que colaboran son listadas a la derecha de cada responsabilidad (Zambrano, 2014).

Las tarjetas CRC trabajan con la técnica de modelado basada en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades. La filosofía del CMS Drupal no está basada en la Programación Orientada a Objetos (POO), por tanto en el presente trabajo se adapta la técnica de las tarjetas CRC a las características de Drupal, el cual sustituye las clases por módulos. Las tarjetas CRC diseñadas en el presente trabajo se realizan con la intención de representar un módulo y no una clase, quedando las responsabilidades definidas como las funciones que estos realizan y los colaboradores son los módulos con los cuales se relacionan. A continuación se muestra la relación entre Módulo-Responsabilidad-Colaboración o tarjeta CRC correspondiente al módulo Automatizado.

Tabla 6: TarjetaCRC_Automatizado.

TarjetaCRC_Automatizado	
Super Clase : Automatizado.	
Descripción: Se encarga del control de la vigencia de las disposiciones jurídicas teniendo en cuenta la fecha de promulgación, la fecha de publicación, la fecha actual y la cantidad de días que va entra en vigor.	
Responsabilidades:	
Colaborador	
Nombre	
Obtener las disposiciones jurídicas que no estén vigentes.	Node
Calcular el tiempo en que puede entrar en vigor una disposición jurídica.	Node
Actualizar el estado de vigencia de la disposición jurídica.	Node

2.4 Patrones de diseño

En este epígrafe se describen cada uno de los patrones empleados para el desarrollo de la aplicación web.

2.4.1 Patrones de diseño de base de datos.

Llaves subrogadas: El uso de este patrón es bastante generalizado en la base de datos de la aplicación web, pues la mayoría de las entidades contienen una llave única entera auto-incremental. Esto facilita reducir el costo de las búsquedas en la base de datos, protección ante los cambios debido a que la lógica del negocio no está en las llaves y evita la contención (bloqueo) de la base de datos, ya que los mecanismos de generación secuencial son rápidos y los provee la aplicación. La Figura 3 muestra el uso de este patrón en la entidad Documento.

Documento		
	iddocumento	integer(10)
	título	integer(10)
	No	varchar(255)
	fecha-publicacion	date
	fecha-promulgacion	date
	vigencia	bit
	uj	integer(10) U
	tiene-denominacion	bit
	denominacion	varchar(255)
	texto	varchar(255)
	<i>idrango</i>	<i>integer(10)</i>
	<i>idpromulgador</i>	<i>integer(10)</i>
	<i>idmateria</i>	<i>integer(10)</i>
	<i>idcharacter</i>	<i>integer(10)</i>
	<i>iddestinatario</i>	<i>integer(10)</i>
	<i>idestado</i>	<i>integer(10)</i>

Figura 3: Empleo del patrón llaves subrogadas en la entidad Documento.

Control de Acceso Basado en Roles (RBAC, por sus siglas en inglés): este patrón se implementa mediante la asignación de roles a los diferentes usuarios que deben tener acceso a la aplicación web, a través de los cuales estos obtienen los permisos para realizar determinadas funciones. Todo esto simplifica la gestión de los usuarios, debido a que el

manejo de los permisos se reduce solo a asignarle el rol apropiado. Es importante resaltar que el patrón RBAC es implementado por el gestor de contenido Drupal.

2.4.2 Patrones implementados en el CMS Drupal.

Drupal utiliza un conjunto de patrones de diseños pertenecientes al “Grupo de los Cuatro”, Gang of Four (GoF), los mismos se describen a continuación (drupal.org, 2015):

Singleton (Solitario o instancia única):

Este patrón se evidencia asociando los módulos y temas como objetos, cada uno de ellos en si no almacena datos. Lo que separa a un módulo de otro es el conjunto de funciones que contiene, por lo que debe ser considerado como una clase con una instancia única.

Decorator (Decorador):

Drupal hace uso de este patrón para extender el comportamiento de todos los nodos, así permite una amplia variedad de comportamientos que se añaden a los nodos sin la necesidad de subclases.

Observer (Observador)

Similar al Decorator, se emplea para extender el comportamiento de los componentes de Drupal a través de sus correspondientes ganchos.

Bridge (Puente)

La capa de abstracción de base de datos de Drupal se implementa de una manera similar a la del patrón Puente. Los módulos deben ser programados de manera que sea independiente del sistema de base de datos que se utiliza, y la capa de abstracción proporciona para esto.

Command (Comando)

Muchos de los hook¹¹ de Drupal utilizan el patrón de Command para reducir el número de funciones que son necesarias implementar, pasando la operación como parámetro junto con los argumentos. Los hooks utilizan este patrón, por lo que los módulos no tienen que definir hook, sino solo los que tienen que ver con su propósito y funcionamiento.

Chain of responsibility (Cadena de responsabilidades)

El sistema de menús de Drupal es la evidencia del patrón Chain of responsibility .En cada petición de una página, el sistema de menús de Drupal determina si hay algún módulo para

¹¹ **Hook:** Consisten en una función PHP utilizada por Drupal que se nombra de la forma <nombre del módulo>_<nombre del gancho>. Cada uno tiene un conjunto definido de parámetros y el tipo de resultado especificado.

responder la petición, si el usuario tiene acceso al recurso solicitado y que función se debe llamar para procesar la petición. En este proceso se trasmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos. De esta forma se continúa la cadena hasta que un módulo atienda la petición, deniegue el acceso o la cadena se agote.

2.4.3 Otros patrones de diseño empleados.

En el diseño de la aplicación web se emplearon además patrones de diseño de asignación de responsabilidades o GRASP (Patrones Generales de Software para Asignar Responsabilidades), los cuales se describen a continuación.

Experto: El uso de este patrón se evidencia en el uso de clases expertas, pues estas contienen la información necesaria para cumplir con las responsabilidades que le fueron asignadas. De este modo la información se mantiene encapsulada y disminuye el acoplamiento. La Figura 4 muestra el uso de este patrón en la clase Materia.

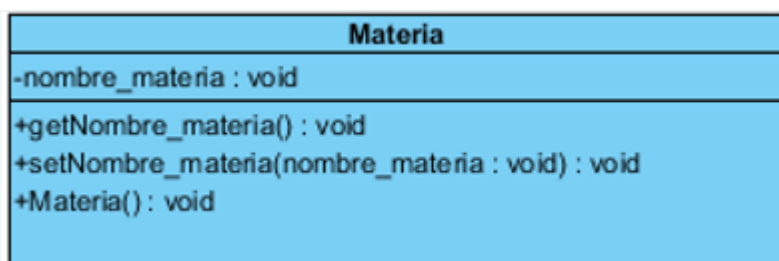


Figura 4: Clase experta en información referente a materias de las disposiciones jurídicas.

Creador: el patrón se evidencia en las clases controladoras que, para cada uno de los módulos o funcionalidades de la aplicación web, son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento. La implementación de este patrón está incluida en la arquitectura de Drupal.

Bajo Acoplamiento: se proporciona un bajo acoplamiento en el diseño debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependan en gran medida de otras, permitiendo de esta forma tener sistemas más robustos y de fácil mantenimiento.

Alta cohesión: el patrón se evidencia debido a que a cada una de las clases se le asignaron responsabilidades de tal forma, que estén estrechamente relacionadas entre sí y elimina la sobrecarga de responsabilidades. La Figura 5 muestra el uso de este patrón en el diseño de las principales clases de la aplicación.

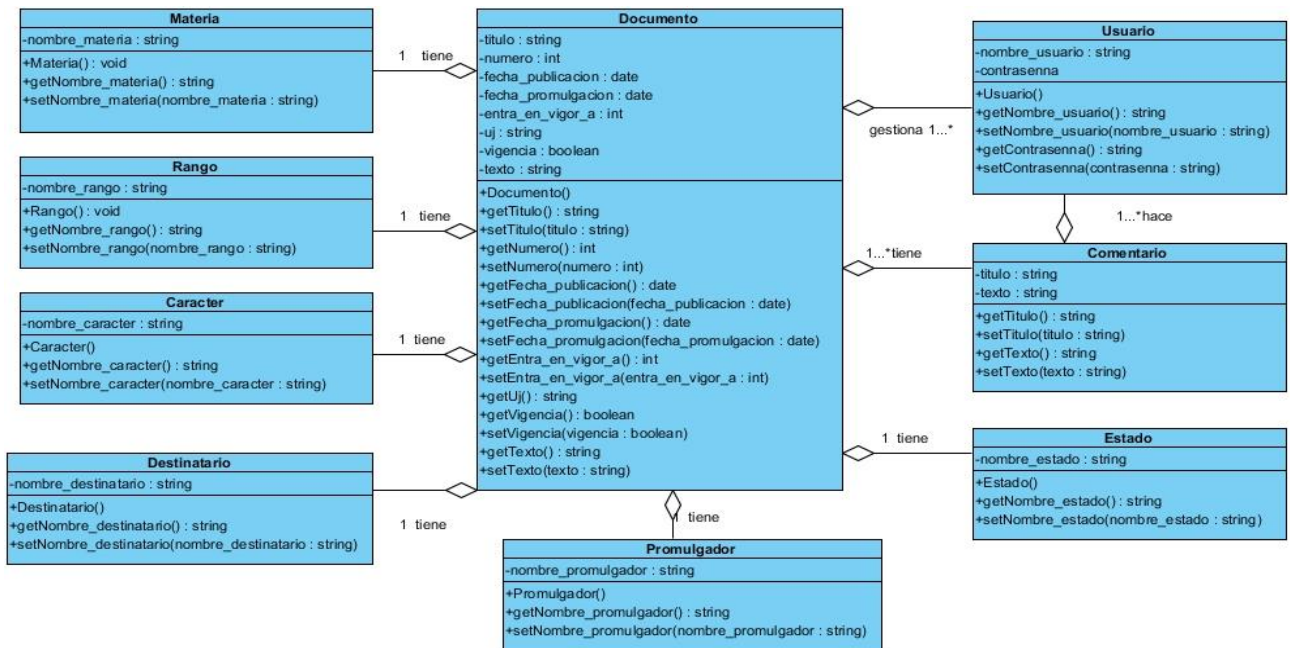


Figura 5: Presencia del patrón Alta Cohesión en la aplicación.

2.5 Descripción de la base de datos

El modelo de datos constituye el conjunto de herramientas conceptuales para describir datos, las relaciones entre los datos, su significado y sus restricciones de consistencia (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) (Proal Aguilar, 2006).

Para el desarrollo de la solución propuesta se realizó un modelo de datos compuesto por un total de 18 entidades. La tabla documentos constituye una de las entidades más importantes en el desarrollo de la solución, así como las entidades rango, carácter, materia, destinatario, promulgador y estado, las cuales tienen una relación de uno a mucho con la tabla documentos. La importancia de estas tablas en el modelo se debe a que en las mismas se almacenan todos los datos correspondientes a las disposiciones jurídicas, actos normativos que constituyen la base para el desarrollo del presente trabajo.

En el modelo también se tuvo en cuenta el diseño de entidades vinculadas a los datos de los usuarios que deben acceder a la aplicación, los roles y permisos que deben poseer y acciones que pueden realizar sobre la tabla documento, ejemplo los comentarios. A continuación en la Figura 3 se muestra el modelo de datos diseñado para la aplicación web obtenida, acotado solo a sus clases y relaciones. En el Anexo 4 Figura 17 se observa el diagrama y sus atributos.

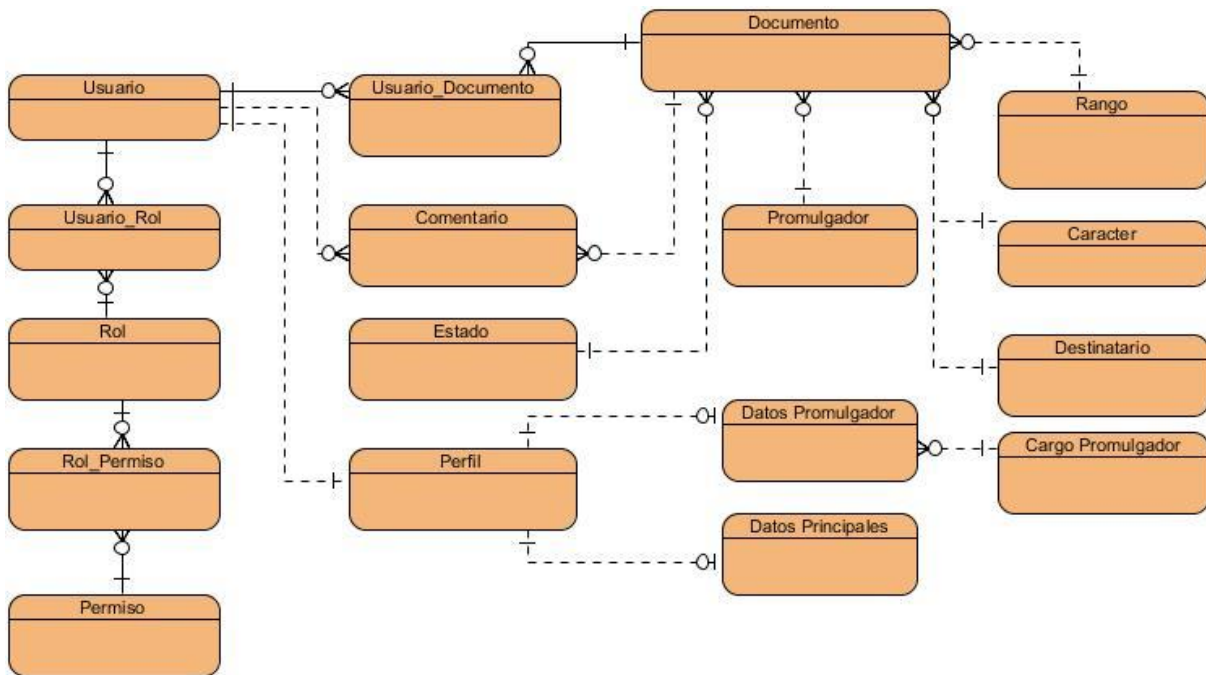


Figura 6: Modelo de datos.

2.6 Conclusiones parciales

- El uso de la metodología XP permitió estructurar y organizar el trabajo en función de las fases que define, generando los artefactos correspondientes, en función de describir todo el proceso de desarrollo de la aplicación web obtenida.
- El empleo del patrón arquitectónico M-V-C permite la reutilización de los componentes definidos y el desarrollo de una solución escalable, de fácil implementación y mantenimiento.
- El empleo de patrones en el diseño de las clases y la base de datos, permitió obtener una solución con poca dependencia entre clases y roles bien definidos que garanticen seguridad en el acceso a la información.

Capítulo 3: Implementación y pruebas de la aplicación web

3.1 Introducción

En el siguiente capítulo se da continuidad a la fase de Iteraciones, con el análisis de la implementación de la aplicación web, así como la descripción del empleo del resto de las fases de la metodología XP. De la implementación se definen las tareas de ingeniería y los estándares de codificación tenidos en cuenta por los desarrolladores. En relación a las fases de Producción y Mantenimiento se representa el diagrama de despliegue y las pruebas realizadas a la aplicación, de estas últimas se representan los resultados de aplicar las métricas y técnicas empleadas en la validación del diseño e implementación de la solución propuesta. Por último se describe cómo a través de la aplicación web quedan resueltas cada una de las deficiencias descritas en la problemática que dio lugar al origen de la presente investigación.

3.2 Implementación

La implementación, como fase del proceso de desarrollo de software, se nutre de los resultados obtenidos en el diseño. En esta fase las HU se descomponen en tareas de programación o ingeniería, que a su vez son convertidas en código. En correspondencia con lo antes descrito se procede a la implementación de la aplicación web que responde al objetivo general de la presente investigación.

3.2.1. Tareas de ingeniería

Una vez identificadas las HU, los programadores proceden a descomponer cada una en tareas específicas, denominadas tareas de la programación o tareas ingenieriles, las cuales están escritas técnicamente y darán solución a cada HU. Las tareas de ingeniería permiten a los desarrolladores obtener un nivel de detalle más avanzado por las HU. A continuación se describe la tarea de ingeniería, para la HU Control de vigencia de disposiciones jurídicas:

Tabla 7: Tarea de ingeniería 2 Control de vigencia de disposiciones jurídicas.

Tarea de ingeniería	
Número de tarea: 3	Historia de Usuario (No.2): Control de vigencia de disposiciones jurídicas
Nombre de tarea: Implementar HU_ Control de vigencia de disposiciones jurídicas	
Tipo de tarea: Desarrollo	Puntos estimados: 3 semanas
Fecha de inicio: 25-05-2015	Fecha de fin: 11-06-2015
Programador responsable: Dailenis Alcántara Frómata	

Descripción: este requisito debe permitir realizar el control de la vigencia de las disposiciones jurídicas teniendo en cuenta la fecha de promulgación, la fecha de publicación, la fecha actual y cantidad de días que entrará en vigor la disposición jurídica.

3.2.2 Estándares de codificación

Para la declaración del código se tuvieron en cuenta los siguientes estándares (Gil, 2012):

- En Drupal la indentación debe ser de 2 espacios.
- Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, \$numero = 3.
- Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, \$numero++.
- Cuando se escriba en PHP, siempre se deben utilizar las etiquetas <?php y ?>, y en ningún caso la versión corta <? y ?>.
- En los ficheros *.module* y *.inc* se omite la etiqueta de cierre de PHP (?>).
- En los archivos de plantilla *.tpl.php*, cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.
- Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.
- La llave de apertura ({) se sitúa en la misma línea que la definición de la estructura, separada por un espacio.
- Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso.
- Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones.
- El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras.
- En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. Ejemplo: `function forum_help($path, $arg) {`.
- Se pueden utilizar las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea, suelen escribirse al principio de un archivo o de cada función. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto.

3.2 Diagrama de despliegue

El Diagrama de despliegue se emplea para visualizar como están relacionados físicamente los componentes de la aplicación. Está generalmente representado por nodos que pueden ser un dispositivo de hardware o algún entorno de ejecución de software. La Figura 7 muestra el diagrama de despliegue definido para la solución propuesta.

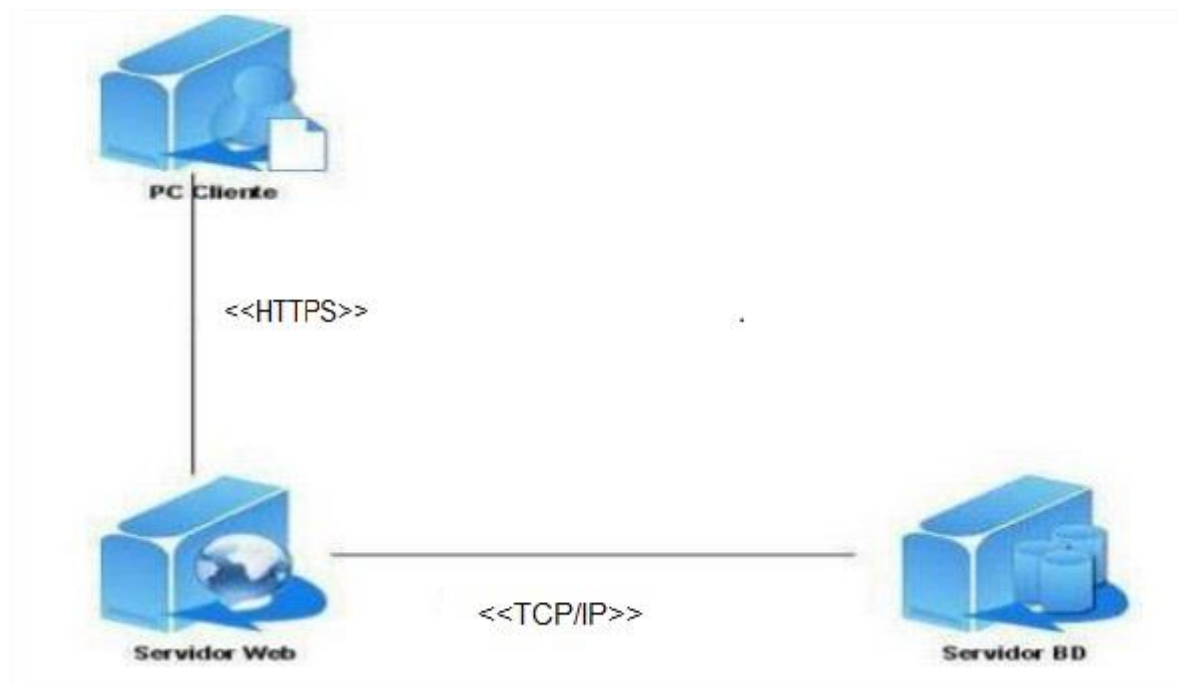


Figura 7: Diagrama de despliegue.

Nodos:

- **PC cliente:** representa una computadora desde la cual el usuario accede a la aplicación. En el caso del escenario para el cual se va a desplegar la aplicación web obtenida, estas PC clientes van a estar ubicadas en las instituciones que constituyen emisores de disposiciones jurídicas hacia el MINJUS y en el propio ministerio.
- **Servidor web:** representa una estación donde se instala el servidor Apache sobre el cual se despliega la aplicación web. Realiza la función de intermediario entre las PC Clientes que realizan sus peticiones y el servidor de BD donde se encuentran las informaciones. En el caso de la solución propuesta en el presente trabajo, el servidor web debe estar montado en el MINJUS.
- **Servidor de base de datos:** representa el servidor donde se instala el Sistema Gestor de Base de Datos MySQL que responde a las peticiones hechas por la aplicación. Este servidor para el caso de la solución obtenida también debe de estar ubicado en el MINJUS.

Protocolos:

- **TCP/IP (Protocolo de comunicación y transmisión/ Protocolo de Internet):** se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.
- **HTTP (Protocolo de transferencia de hipertexto):** establece un esquema de comunicación cliente–servidor. El cliente es el navegador web instalado en las PC clientes, quien realiza las peticiones al servidor y este se encarga de dar respuesta

3.3 Técnicas de validación de los requisitos

Con el objetivo de demostrar que los requisitos de software obtenidos cumplen con las expectativas que el cliente desea, se llevó a cabo un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

- **Revisiones formales de los requisitos:** Se realizaron revisiones formales a los requisitos por parte del equipo de desarrollo y por el cliente. Las revisiones por parte del equipo de desarrollo no generaron no conformidades, en cambio con el cliente se realizaron 2 revisiones. En la primera se modificaron 2 RF, y en la segunda revisión el cliente quedó satisfecho con el trabajo realizado por el equipo de desarrollo y se aprobaron los requisitos, generándose de este encuentro un Acta de Aceptación por parte del cliente (Ver Anexo 1 Figura 14).
- **Generación de casos de prueba (CP):** como parte del proceso de validación de los requisitos funcionales del sistema se diseñaron un conjunto de CP para cada HU definida.

Al concluir el proceso de validación de requisitos con la aplicación de las técnicas antes descritas se demostró que los RF definidos para el desarrollo de la aplicación web están en correspondencia con las necesidades del cliente.

3.4 Validación del diseño

Para comprobar la calidad del diseño de la aplicación web se emplearon las métricas Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC).

La métrica TOC permite medir la responsabilidad, la complejidad de implementación y la reutilización de las clases del diseño (Pressman, 2002). Es importante destacar que para esta métrica, la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que a mayor responsabilidad y complejidad de implementación de una clase, menor será su nivel de reutilización.

El tamaño operacional de una clase se puede determinar sumando todas las operaciones que posee, el resultado es tomado como umbral. El próximo paso es calcular el promedio correspondiente a los umbrales, una vez obtenidos se procede a calcular el comportamiento de cada clase respecto a los atributos de calidad establecidos, de acuerdo a lo expuesto en la siguiente tabla:

Tabla 8: Rango de valores para medir la afectación de los atributos de calidad (TOC).

Atributos de calidad	Clasificación	Criterio
Responsabilidad	Baja	Umbral <= Promedio
	Media	Promedio < Umbral <= 2* promedio
	Alta	Umbral > 2* promedio
Complejidad de implementación	Baja	Umbral <= Promedio
	Media	Promedio < Umbral <= 2* promedio
	Alta	Umbral > 2* promedio
Reutilización	Baja	Umbral > 2* promedio
	Media	Promedio < Umbral <= 2* promedio
	Alta	Umbral <= Promedio

En la Figura 8 se muestran los resultados obtenidos de la aplicación de esta métrica al diseño de la aplicación web:

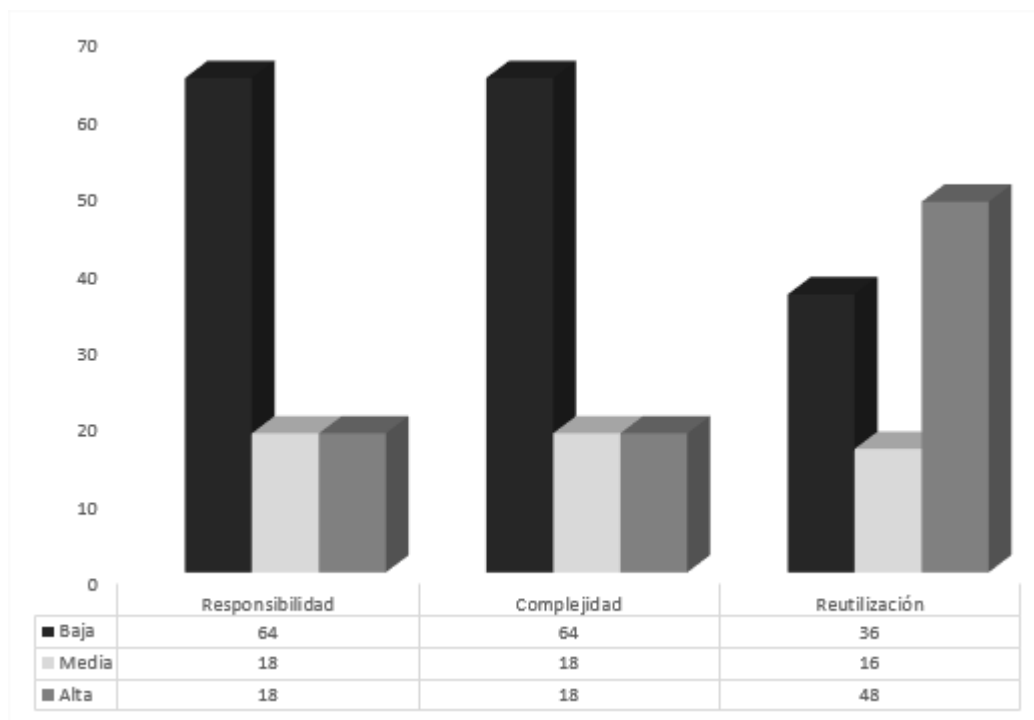


Figura 8: Representación en (%) de los resultados de la aplicación de la métrica TOC.

Una vez aplicada la métrica, se observa que el diseño de la aplicación presenta bajo nivel de responsabilidad y complejidad, garantizando en gran medida la reutilización. De esta manera favorecen el bajo acoplamiento y la alta cohesión.

La métrica RC permite evaluar el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar una clase del diseño, teniendo en cuenta las relaciones existentes entre ellas. En este caso se mide la relación entre el modulo implementado y los módulos personalizados.

Para determinar el grado de afectación para los atributos de calidad que mide la métrica RC es necesario conocer la cantidad de relaciones de uso (CRU) que poseen las clases a medir. Una vez obtenida la CRU, se procede a calcular el promedio de las mismas. Teniendo en cuenta ambos valores, según los criterios expuestos en la siguiente tabla, se determina la incidencia de los atributos de calidad en cada una de las clases.

Tabla 9: Rango de valores para medir la afectación de los atributos de calidad (RC).

Atributos de calidad	Clasificación	Criterio
Acoplamiento	Ninguna	CRU = 0
	Baja	CRU = 1
	Media	CRU = 2
	Alta	CRU > 2
Complejidad de mantenimiento	Baja	CRU <= Promedio
	Media	Promedio < CRU <= 2* promedio
	Alta	CRU > 2* promedio
Reutilización	Baja	CRU > 2* promedio
	Media	Promedio < CRU <= 2* promedio
	Alta	CRU <= Promedio
Cantidad de pruebas	Baja	CRU <= Promedio
	Media	Promedio < CRU <= 2* promedio
	Alta	CRU > 2* promedio

En la Figura 9 se muestran los resultados obtenidos de la aplicación de esta métrica al diseño de la aplicación web:

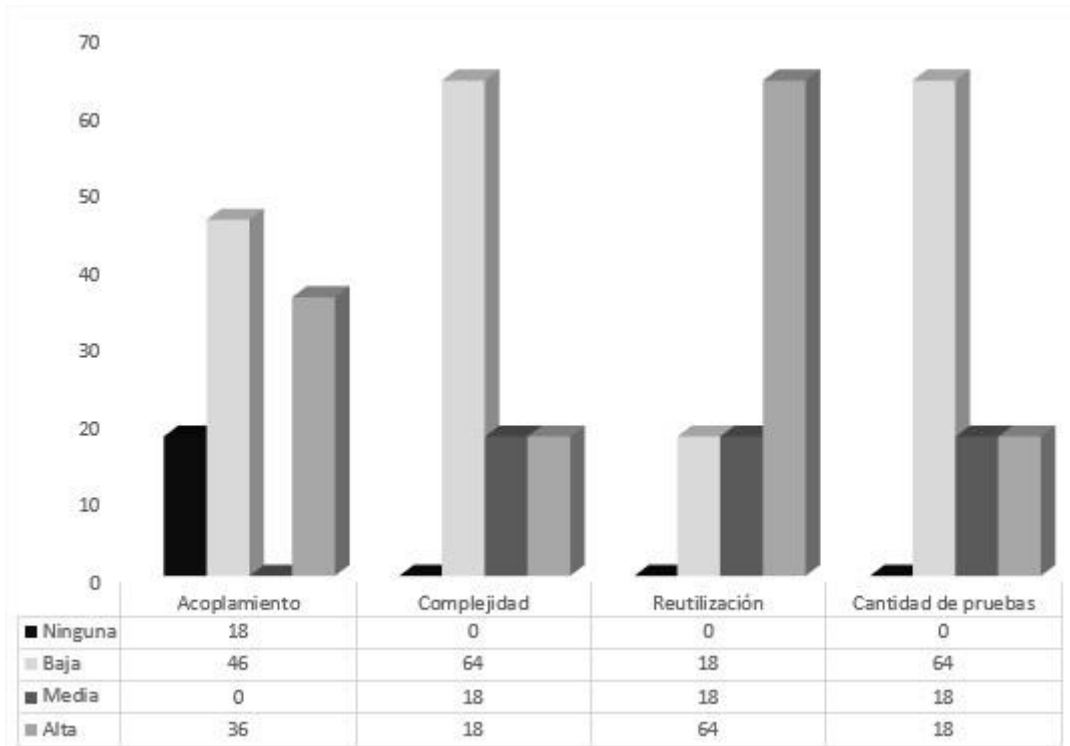


Figura 9: Representación en (%) de los resultados de la aplicación de la métrica RC.

Una vez aplicada la métrica RC se obtiene como resultado que el diseño de la aplicación web promueve el bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas no son altas y en consecuencia el grado de reutilización es mayor.

De la aplicación de las métricas TOC y RC se obtuvo como resultado que el sistema no presenta un diseño complejo, que los módulos presentan bajo acoplamiento, una alta cohesión y el grado de reutilización es alto.

3.5 Pruebas

Durante la implementación del software las posibilidades de cometer errores son múltiples por lo que se hace necesario realizar pruebas que contribuyan a detectar las irregularidades del software. Las pruebas permiten comprobar el grado de cumplimiento de las especificaciones que se implementan en el sistema. Existen diferentes tipos de pruebas, entre ellas se destacan las pruebas unitarias y las de aceptación.

3.5.1 Pruebas unitarias

Las pruebas unitarias están enfocadas al código fuente de los componentes para verificar todos los flujos de control, probando de manera individual las partes del sistema que han sido desarrolladas. Las pruebas unitarias se derivan en pruebas funcionales (caja negra) y pruebas estructurales (caja blanca).

3.5.1.1 Pruebas de caja blanca

Las pruebas de Caja Blanca son las que se llevan a cabo en la parte interna del software, específicamente sobre el código fuente. Se basan en el examen minucioso de los detalles procedimentales. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles (Pressman, 2002).

Las pruebas de caja blanca se realizaron mediante el empleo de la técnica del Camino Básico. Esta técnica permitió obtener una medida de complejidad lógica y con esta medida se generó un conjunto de caminos básicos que se ejecutan por lo menos una vez durante la ejecución del programa, los cuales son utilizados para obtener los casos de prueba. En la Figura 10 se muestra uno de los métodos a los cuales se le aplica la técnica de camino básico. Esta porción de código representa la función *automatizado_cron*, que responde a la funcionalidad Control de vigencia, una de las más importantes para el cumplimiento del objetivo general de la presente investigación.

```
function automatizado_cron() {
    $documentos_nids = db_query("SELECT nid
        FROM node n
        INNER JOIN field_data_field_fecha_de_promulgacion f_pro ON n.nid = f_pro.entity_id
        INNER JOIN field_data_field_fecha_de_publicacion f_pub ON n.nid = f_pub.entity_id
        INNER JOIN field_data_field_entra_en_vigor_a f_e_v ON n.nid = f_e_v.entity_id
        INNER JOIN field_data_field_vigente v ON n.nid = v.entity_id
        WHERE n.type = 'documento'
        AND DATEDIFF(NOW(), f_pub.field_fecha_de_publicacion_value) >= f_e_v.field_entra_en_vigor_a_value
        AND v.field_vigente_value = 0")->fetchCol();

    if (count($documentos_nids)) {
        $nodes = node_load_multiple($documentos_nids);

        foreach ($nodes as $node) {
            $node->field_vigente['und'][0]['value'] = 1;
            node_save($node);
        }
    }
}
```

Figura 10: Función Automatizado_cron.

Para aplicar la técnica del camino básico se realizaron una serie de pasos, los cuales se describen a continuación:

1. **Confeccionar el grafo de flujo:** usando el código de la Figura 10 se realizó la representación del grafo de flujo, el cual está compuesto por los siguientes elementos:
 - Nodos: son círculos que representan una o más sentencias procedimentales.
 - Aristas: son flechas que representan el flujo de control y son análogas a las flechas del diagrama de flujo.
 - Regiones: son las áreas delimitadas por aristas y nodos.

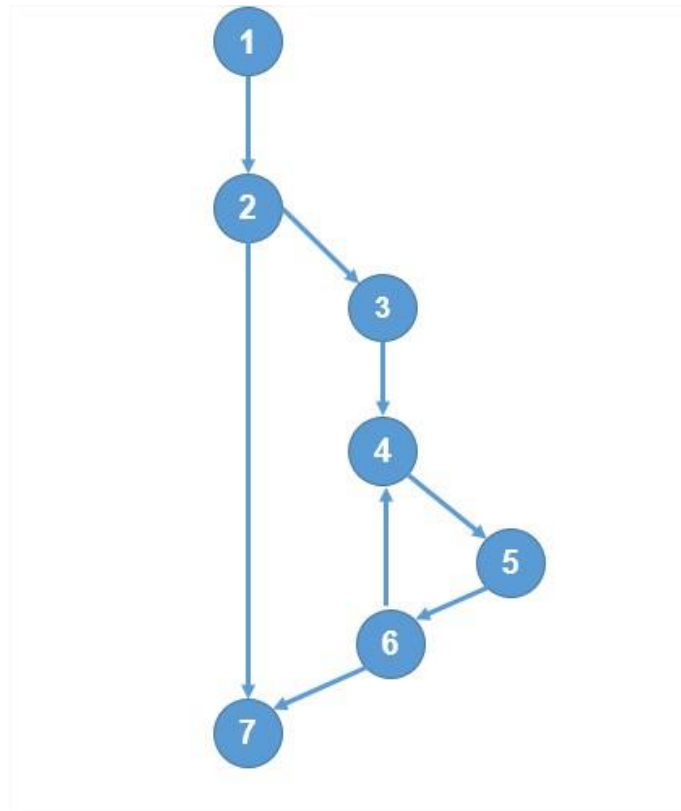


Figura 11: Grafo del camino básico de la función Automatizado_cron.

2. **Calcular la complejidad ciclomática:** proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa y da un límite inferior para el número de CP que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática se calcula de tres formas:

- El número de regiones del grafo de flujo.

$$V(G) = 3$$

- $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

$$V(G) = A - N + 2 = 8 - 7 + 2 = 3$$

- $V(G) = P + 1$, donde P es el número de nodos predicados (representan una condicional en el grafo de flujo).

$$V(G) = P + 1 = 2 + 1 = 3$$

3. **Determinar un conjunto básico de caminos linealmente independientes:** el valor de $V(G)$ da el número de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los 3 caminos siguientes:

Camino básico #1: 1-2-7

Camino básico #2: 1-2-3-4-5-6-7

Camino básico #3: 1-2-3-4-5-6-4-5-6-7

4. **Obtención de CP:** cada camino independiente es un caso de prueba a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino. En este caso se obtuvieron 3 caminos básicos, por tanto se hace necesario la confección de igual número de CP, para aplicar las pruebas a este método. A continuación se brinda un ejemplo de uno de ellos:

Tabla 10: Caso de prueba de caja blanca para el camino básico #2.

Caso de prueba: Camino básico #4	
Entrada	Disposición jurídica
Resultados esperados	Actualizar el campo “vigente” de la disposicione jurídica que entran en vigor.
Condiciones	DATEDIFF(NOW(), f_pub.field_fecha_de_publicacion_value) >= f_e_v.field Entra en vigor a_value, v.field_vigente_value = ,0

Una vez aplicadas las pruebas de caja blanca mediante la técnica del camino básico a la función `Automatizado_cron`, se detectó en una primera iteración un error en el código. Dicho error fue erradicado y se procedió a aplicar una segunda iteración de estas pruebas, de las cuales se obtuvieron resultados satisfactorios.

De igual forma se le realizaron estas pruebas al resto de las funcionalidades de la aplicación web, en un total de dos iteraciones. A continuación la Figura 12 muestran los resultados obtenidos:



Figura 12: Resultados obtenidos de la aplicación de la técnica del camino básico.

3.5.1.2 Pruebas de caja negra

Las pruebas de caja negra son pruebas que se realizan a las funcionalidades del software. Para la realización de las mismas se elaboraron un conjunto de CP, que constituyen una guía para comprobar el correcto funcionamiento de la aplicación web obtenida. Los objetivos de practicar estas pruebas son: encontrar errores de funciones incorrectas o ausentes, de interfaz o en accesos a la base de datos. A continuación se describe el CP diseñado para el RF10: Registrar disposición jurídica.

Tabla 11: Caso de prueba de caja negra de la funcionalidad Registrar disposición jurídica.

Escenario	Descripción	Documento	Respuesta del sistema
EC 1.1 Introducir datos correctos.	Introducir los datos de una disposición jurídica.	V	Registrar la disposición jurídica.
		Datos correctos.	
EC 1.2 Campos vacíos.	Dejar campos vacíos de una disposición jurídica.	F	No se registra la disposición jurídica y muestra un mensaje de error
		Campos vacíos obligatorios.	
EC 1.3 Introducir datos incorrectos.	Introducir datos incorrectos de una disposición jurídica.	F	No se registra la disposición jurídica y muestra un mensaje de error.
		Datos incorrectos	

Para comprobar que cada una de las funcionalidades se ejecuta correctamente y responden a las necesidades del cliente, la aplicación web fue revisada por el grupo de Calidad CEGEL, aplicando cada uno de los CP diseñados. El desarrollo de las pruebas se realizó en un total de 3 iteraciones.

A continuación la Figura 13 ilustra el resultado de aplicar las mismas, teniendo en cuenta los parámetros de interfaz, redacción, recomendación y ortografía.

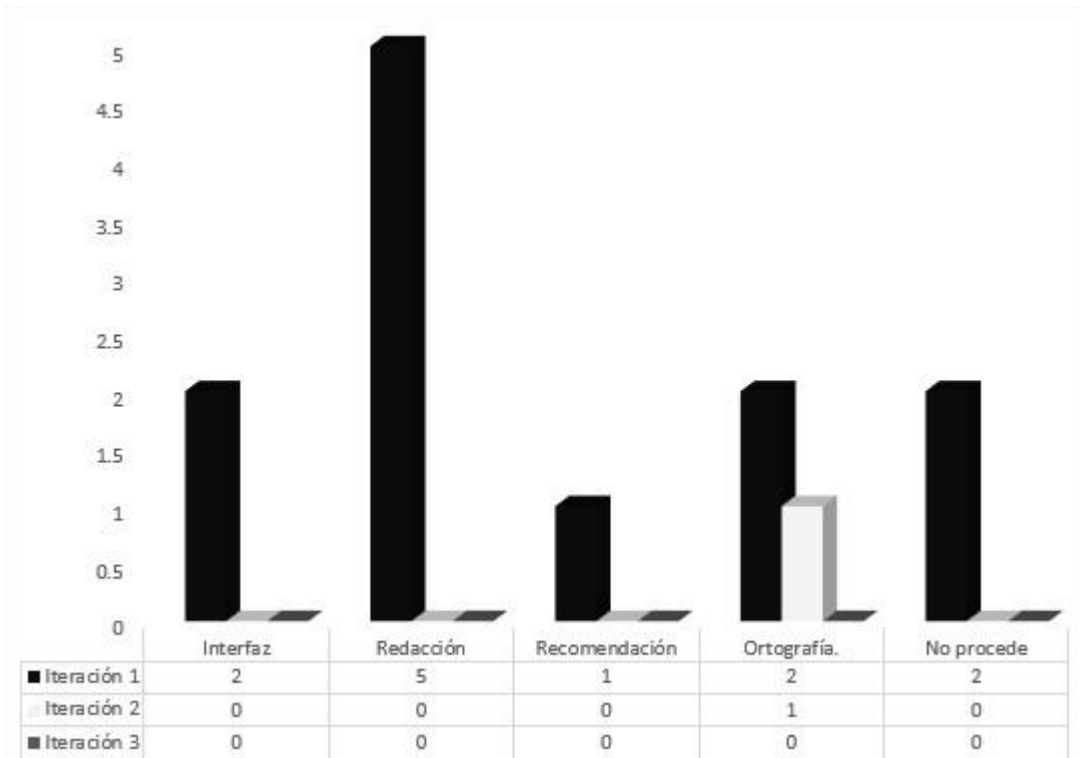


Figura 13: No conformidades detectadas en la aplicación a través de las pruebas de caja negra.

En una primera iteración se detectaron un total de 12 No Conformidades (NC), predominando entre ellas las de tipo redacción. De estas NC, 10 se resolvieron y 2 no procedieron. En la segunda iteración se detectó 1 NC de tipo ortografía, la cual fue resuelta. Luego en una tercera iteración se obtuvieron resultados satisfactorios, debido a que no se encontraron NC, generándose el Acta de Liberación Interna de Productos de Software (Ver Anexo 2 Figura 15).

3.5.2 Pruebas de aceptación

Las pruebas de aceptación, constituyen una técnica para garantizar que los requisitos hayan sido cumplidos y que la solución obtenida es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento. Estas son creadas a partir de las HU y desde la perspectiva del cliente. El objetivo final es lograr que los requerimientos sean cumplidos y que la aplicación web sea aceptable. Tal es así, que una vez que todas las HU hayan pasado sus pruebas de aceptación se considera entonces terminado el trabajo.

Para la aplicación de estas pruebas se confeccionaron CP de aceptación por cada HU. Seguidamente se muestra el caso de prueba de aceptación de la HU “Control de vigencia de disposición jurídica”.

Tabla 12: Caso de prueba de Aceptación de la HU “Control de vigencia de la disposición jurídica”.

Caso de prueba de aceptación		
Código de caso de prueba: 07		Nombre historia de usuario: Control de vigencia de la disposición jurídica.
Nombre de la persona que realiza el caso de prueba: Dailenis Alcántara Frómeta.		
Descripción de la prueba: revisar a través de la aplicación web el correcto funcionamiento del RF 14: Control de vigencia de la disposición jurídica.		
Condiciones de ejecución: debe de existir una disposición jurídica insertada.		
Entrada/Pasos de ejecución		Resultados esperados:
Acción:	Entrada:	
Se selecciona la opción: “Ver contenido”	Texto.	
El sistema debe mostrar el campo vigente actualizado.		
Evaluación de prueba: Satisfactoria		

Se realizó un encuentro con la MSc. Yarina Amoroso Fernández, presidenta de la Sociedad Cubana de Derecho e Informática, con el objetivo de revisar la aplicación web, teniendo en cuenta los CP definidos. Los resultados obtenidos fueron satisfactorios, avalados por la especialista entrevistada como un aporte importante para la informatización de los procesos en el MINJUS. Al finalizar el encuentro se generó el Acta de Aceptación del Producto (Ver Anexo 3 Figura 16).

3.6 Aporte de la investigación

En la actualidad la inscripción y revisión técnica de los documentos asentados en el Registro de Disposiciones Jurídica del MINJUS se realiza de forma manual. El proceso desde que un organismo emisor envía una disposición jurídica al MINJUS y esta es revisada por un especialista, para luego pasar a edición y ser publicada en el Gaceta Oficial de la República, tarda como mínimo 15 días. Además las disposiciones se trasladan desde los organismos hasta el ministerio en formato duro y por mediación de una persona responsable de transportarlas.

Con la puesta en práctica de la aplicación web obtenida se contribuye a agilizar el desarrollo del proceso antes descrito. El uso de la misma evita el traslado de las disposiciones jurídicas en formato papel, pues con solo existir una conexión a través de la web desde cada organismo emisor hacia el MINJUS, donde debe estar instalado el servidor de la aplicación, se garantizan que los especialistas de cada organismo puedan subir por la web cada disposición jurídica generada. Esto trae como

resultado que los encargados de su revisión en el ministerio puedan tener acceso de inmediato a la misma y mantener retroalimentados a los emisores sobre el proceso de revisión técnica de cada disposición recibida.

Otras de las contribuciones de la solución informática obtenida es el ahorro de recursos materiales y de esfuerzo de horas-hombres en el trabajo, al mejorar la gestión del proceso de inscripción y revisión técnica de las disposiciones jurídicas. Esto se justifica teniendo en cuenta que a partir de ahora las disposiciones no se enviarán de forma manual y en formato papel desde el organismo emisor hasta el MINJUS, eliminando así el gasto de papel y el empleo de las personas encargadas de llevar cada acto normativo hacia el ministerio. Por otra parte con la aplicación web se logra tener un control de la vigencia de las disposiciones jurídicas publicadas en la Gaceta Oficial de la República de Cuba, pues la misma controla la fecha de publicación y el tiempo en que demorara para entrar en vigor cada rango jurídico publicado, proceso que hoy en día no se controla.

3.7 Conclusiones parciales

- El empleo de los estándares de codificación seleccionados y el uso de un conjunto de convenciones de nomenclatura, contribuyeron al entendimiento y estandarización del código de la solución obtenida, facilitando el futuro mantenimiento de la misma.
- La descripción del diagrama de despliegue permitió obtener una aproximación de los elementos a tener en cuenta para la futura implantación en un escenario real de la solución obtenida.
- La aplicación de las técnicas de validación de requisitos: Revisiones formales de los requisitos y Generación de CP, permitieron obtener requerimientos entendibles y acordes a las necesidades del cliente, avalados por el acta de aceptación de requisitos obtenida al final del proceso.
- Con la validación del diseño de la herramienta obtenida mediante la aplicación de las métricas TOC y RC, se obtuvo como resultado que las clases presentan bajo acoplamiento y poseen un alto grado de reutilización.
- La realización de pruebas unitarias y de aceptación a la aplicación web obtenida permitió comprobar el correcto funcionamiento de la misma.

Conclusiones Generales

- El estudio de soluciones existentes en la actualidad para la publicación de leyes, resoluciones, circulares y demás actos normativos en países de Latinoamérica, demostró que los trabajos realizados no dan respuesta a la problemática que dio origen a la presente investigación, fundamentando la necesidad de desarrollar la aplicación web propuesta.
- La utilización de la metodología XP y el uso del patrón arquitectónico Modelo-Vista-Controlador en la implementación de la aplicación web obtenida, posibilitó agilizar el desarrollo de la misma y obtener una solución escalable y de fácil mantenimiento.
- El uso de técnicas de validación de requisitos, métricas de validación del diseño y la aplicación de pruebas unitarias y de aceptación, permitieron obtener una aplicación web funcional estable, avalada en cada caso por las actas de liberación y aceptación obtenidas.
- El resultado del trabajo constituye un aporte en materia de informatización para el Ministerio de Justicia de la República de Cuba, pues la aplicación web desarrollada permite realizar los procesos de inscripción, revisión técnica y control de vigencia de disposiciones jurídica.

Recomendaciones

- Integrar la aplicación con el portal web de la Gaceta Oficial de la República de Cuba.
- Incorporar a la aplicación nuevas funcionalidades que permitan ampliar la informatización de los procesos jurídicos desarrollados en el Ministerio de Justicia de la República de Cuba.

Bibliografía Referenciada

Olson , Philip . 2015. *Manual de PHP.* 2015.

Aguiar, Mayelin. 2003. *Las publicaciones electrónicas: una revolución en el siglo XXI.* Ciudad de La Habana : s.n., 2003. ISBN 1024-9435.

Alfresco. 2013. Alfresco. [En línea] 2013. [Citado el: 21 de febrero de 2015.]
<http://www.alfresco.com/es/>.

Amoroso, Yarina. 2000.. *Informática Jurídica: algunas consideraciones sobre la publicidad de las normas jurídicas.* *Memorias de la III Conferencia Internacional de Derecho en La Habana.* 2000.

—. **2002.** *Redes de Información y Gestión Jurídica. Proyecto de Edición Electrónica de Gaceta Oficial de la República de Cuba.* La Habana : s.n., 2002.

Andujar, José María y Sánchez Artola, Rafael. 2013. Programación extrema o Extreme programming. [En línea] 2013. <http://es.slideshare.net/JoseMariaAndujar/la-programacin-extrema-o-e-xtreme-programming>.

areatecnologia.com. 2015. LENGUAJES DE PROGRAMACION. *areatecnologia.com.* [En línea] 2015.
<http://www.areatecnologia.com/informatica/lenguajes-de-programacion.html>.

boletinoficial.gov.ar. 2014. boletinoficial.gov.ar. *boletinoficial.gov.ar.* [En línea] 2014. [Citado el: 9 de diciembre de 2014.] <http://www.boletinoficial.gov.ar/Inicio/Index.castle>.

Casillas, Luis Alberto, Gibert Ginesta, Marc y Pérez, Oscar. 2013. Base de datos en MySQL. [En línea] 2013. [Citado el: 8 de febrero de 2015.] http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf.

Constitución de la República de Cuba. 2013. *Constitución de la República de Cuba.* La Habana : s.n., 2013.

del Castillo, Alvaro. 2000. *El servidor de web Apache: Introducción práctica.* 2000.

developer.mozilla. 2014. developer.mozilla.org. [En línea] 2014. [Citado el: 7 de febrero de 2015.]
<https://developer.mozilla.org/es/docs/Web/JavaScript>.

2000. *Diccionario de Jurisprudencia.* 2000.

Drupal. 2005. Drupal. [En línea] 2005. [Citado el: 5 de febrero de 2015.]
<http://drupal.org/es/caracteristicas>.

drupal.org. 2015. drupal.org. [En línea] abril de 2015. [Citado el: 8 de junio de 2015.]
<https://www.drupal.org/node/547518>.

Ecured. 2009. Ecured. [En línea] 2009. [Citado el: 5 de diciembre de 2014.]
<http://www.ecured.cu/index.php/SQLite>.

—. Wordpress. [En línea] [Citado el: 4 de febrero de 2015.]
<http://www.ecured.cu/index.php/Wordpress>.

BIBLIOGRAFÍA REFERENCIADA

gacetaoficialdebolivia.gob.bo. 2014. gacetaoficialdebolivia.gob.bo. *Gaceta Oficial del Estado Plurinacional de Bolivia*. [En línea] 2014. [Citado el: 9 de diciembre de 2015.] <http://www.gacetaoficialdebolivia.gob.bo/>.

gacetaoficialdelarepublicabolivarianadevenezuela.com. 2014. gacetaoficialdelarepublicabolivarianadevenezuela.com. *gacetaoficialdelarepublicabolivarianadevenezuela.com*. [En línea] 2014. [Citado el: 9 de diciembre de 2015.] <http://www.gacetaoficialdelarepublicabolivarianadevenezuela.com/2014/>.

Gutiérrez, Javier J. *lsi.us.es. ¿Qué es un framework Web?* [En línea] [Citado el: 23 de mayo de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.

Hechavarría, Angela. 1997. *Las publicaciones Electrónicas: Un concepto, una clasificación y análisis de su impacto en los profesionales de la información*. Centro de Estudios y Desarrollo Profesional en Ciencias de la Información. La Habana : s.n., 1997.

Highsmith, J. Agile. 2002. *Software Development Ecosystem*. s.l. : Addison-Wesley Professional, 2002.

Inforolot. Inforolot. *Sistema de gestión documental*. [En línea] [Citado el: 17 de febrero de 2015.] <http://www.inforolot.com/index.php/es/sistema-de-gestion-documental.html>.

Ivonet, Amarilys de la C. y Hernández Martínez, Ledier. 2014. *Sistema para la edición y publicación electrónica Gaceta Oficial de la República de Cuba*. La Habana : s.n., 2014.

Joomla. 2005. Joomla. [En línea] 2005. [Citado el: 5 de febrero de 2015.] <http://www.joomla.org/about-joomla.html>.

JoomlaSpanish. 2005. JoomlaSpanish. [En línea] 2005. [Citado el: 5 de febrero de 2015.] <http://www.joomlaspanish.org/>.

Menéndez-Barzanallana, Rafael. 2006. *Metodologías de desarrollo de software*. [En línea] 2006. <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.

Meza, Mirna. 2011. *Herramienta CASE*. [En línea] abril de 2011. [Citado el: 7 de febrero de 2015.] <http://fds-herramientascase.blogspot.com/>.

Netbeans.org. 2015. Netbeans.org. *Netbeans.org*. [En línea] 2015. [Citado el: 6 de febrero de 2015.] <https://netbeans.org/features/index.html>.

Orallo, Enrique H. 2012. *El Lenguaje Unificado (UML)*. [En línea] 2012. http://www.acta.es/medios/articulos/informatica_y_computacion/026067.pdf. 026.

Osuna, Maria Rosario y De La Cruz Gómez, Estefanía. 2010. *Los sistemas de gestión de contenido en Información y Documentación*. Salamanca : s.n., 2010.

Pedraza, Alejandro Miguel y González Olivera, Gabriel Alejandro. 2014. *Servicio de asistencia al proceso de consulta popular en la Facultad de Derecho de la Universidad de La Habana*. La Habana : s.n., 2014.

Penadés, Carmen, Letelier, Patricio y Canós, Jase. 2003. *Metodologías Ágiles en el Desarrollo de Software. Metodologías Ágiles en el Desarrollo de Software*. Alicante : s.n., 2003.

BIBLIOGRAFÍA REFERENCIADA

- Postgresql.org. 2010.** Postgresql.org. [En línea] 2010. [Citado el: 5 de diciembre de 2014.] http://www.postgresql.org.es/sobre_postgresql.
- Pressman, Roger. 2002.** *Ingeniería de software*. 2002.
- Priolo, Sebastián. 2007.** Programación Extrema. http://www.fcad.uner.edu.ar/jai/6JAI/XP_6JAI.pdf. [En línea] 2007. [Citado el: 6 de diciembre de 2014.]
- Proal Aguilar, Carlos. 2006.** Modelado de datos. [En línea] 2006. [Citado el: 8 de junio de 2015.] <http://ict.udlap.mx/people/carlos/is341/bases02.html>.
- registroficial.gob.ec. 2014.** registroficial.gob.ec. *registroficial.gob.ec*. [En línea] 2014. [Citado el: 10 de diciembre de 2015.] <https://www.registroficial.gob.ec/>.
- Rodríguez, F.G. 2012.** Experto en Drupal 7. Nivel avanzado. s.l. : Forcontu, 2012.
- Ruiz, Alfredo. 2014.** *¿Cuáles son las características principales del HTML?* 2014.
- Sarduy, Yanetsys y Urra, Pedro. 2006.** *Sistemas de gestión de contenidos: En busca de una e una plataforma ideal*. 2006.
- Slideshare.net. 2012.** Slideshare.net. [En línea] 2012. [Citado el: 5 de diciembre de 2014.] <http://es.slideshare.net/>.
- SQLite. 2015.** <http://www.sqlite.org/>. [En línea] 2015. [Citado el: 7 de febrero de 2015.]
- SQLITE. 2015.** SQLITE. [En línea] 2015. [Citado el: 5 de febrero de 2015.] <http://www.sqlite.org>.
- Tocto, Esteban . 2011.** *Optimización y cuantificación utilizando BPMN*. 2011.
- Trujillo , Yadelis y Fernández, Radamés. 2013.** *Subsistema Mensajería SWIFT Fase II*. La Habana : s.n., 2013.
- w3c.es. 2014.** w3c.es. [En línea] 2014. [Citado el: 7 de febrero de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
- wordpress.org. 2015.** wordpress.org. [En línea] 2015. [Citado el: 10 de febrero de 2015.] <https://wordpress.org/about/>.
- Zambrano, Omar . 2014.** Qué son y para qué sirven las tarjetas CRCs. *scribd.com*. [En línea] 2014. [Citado el: 5 de junio de 2015.] <http://es.scribd.com/doc/207429482/Que-son-y-para-que-sirven-las-tarjetas-CRCs-docx#scribd>.

Bibliografía Consultada

Olson , Philip . 2015. *Manual de PHP.* 2015.

Aguiar, Mayelin. 2003. *Las publicaciones electrónicas: una revolución en el siglo XXI.* Ciudad de La Habana : s.n., 2003. ISBN 1024-9435.

Alfresco. 2013. Alfresco. [En línea] 2013. [Citado el: 21 de febrero de 2015.]
<http://www.alfresco.com/es/>.

Amoroso, Yarina. 2000.. *Informática Jurídica: algunas consideraciones sobre la publicidad de las normas jurídicas.*Memorias de la III Conferencia Internacional de Derecho en La Habana. 2000.

—. **2002.** *Redes de Información y Gestión Jurídica. Proyecto de Edición Electrónica de Gaceta Oficial de la República de Cuba.* La Habana : s.n., 2002.

Andujar, José María y Sánchez Artola, Rafael. 2013. Programación extrema o Extreme programming. [En línea] 2013. <http://es.slideshare.net/JoseMariaAndujar/la-programacin-extrema-o-e-xtreme-programming>.

areatecnologia.com. 2015. LENGUAJES DE PROGRAMACION. *areatecnologia.com.* [En línea] 2015.
<http://www.areatecnologia.com/informatica/lenguajes-de-programacion.html>.

boletinoficial.gov.ar. 2014. boletinoficial.gov.ar. *boletinoficial.gov.ar.* [En línea] 2014. [Citado el: 9 de diciembre de 2014.] <http://www.boletinoficial.gov.ar/Inicio/Index.castle>.

Casillas, Luis Alberto, Gibert Ginesta, Marc y Pérez, Oscar. 2013. Base de datos en MySQL. [En línea] 2013. [Citado el: 8 de Febrero de 2015.] http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf.

Constitución de la República de Cuba. 2013. *Constitución de la República de Cuba.* La Habana : s.n., 2013.

del Castillo, Alvaro. 2000. *El servidor de web Apache: Introducción práctica.* 2000.

developer.mozilla. 2014. developer.mozilla.org. [En línea] 2014. [Citado el: 7 de febrero de 2015.]
<https://developer.mozilla.org/es/docs/Web/JavaScript>.

2000. *Diccionario de Jurisprudencia.* 2000.

Drupal. 2005. Drupal. [En línea] 2005. [Citado el: 5 de febrero de 2015.]
<http://drupal.org/es/caracteristicas>.

drupal.org. 2015. drupal.org. [En línea] abril de 2015. [Citado el: 8 de junio de 2015.]
<https://www.drupal.org/node/547518>.

Ecured. 2009. Ecured. [En línea] 2009. [Citado el: 5 de diciembre de 2014.]
<http://www.ecured.cu/index.php/SQLite>.

—. Wordpress. [En línea] [Citado el: 4 de febrero de 2015.]
<http://www.ecured.cu/index.php/Wordpress>.

BIBLIOGRAFÍA CONSULTADA

gacetaoficialdebolivia.gob.bo. 2014. gacetaoficialdebolivia.gob.bo. *Gaceta Oficial del Estado Plurinacional de Bolivia*. [En línea] 2014. [Citado el: 9 de diciembre de 2015.] <http://www.gacetaoficialdebolivia.gob.bo/>.

gacetaoficialdelarepublicabolivarianadevenezuela.com. 2014. gacetaoficialdelarepublicabolivarianadevenezuela.com. *gacetaoficialdelarepublicabolivarianadevenezuela.com*. [En línea] 2014. [Citado el: 9 de diciembre de 2015.] <http://www.gacetaoficialdelarepublicabolivarianadevenezuela.com/2014/>.

Gutiérrez, Javier J. lsi.us.es. *¿Qué es un framework Web?* [En línea] [Citado el: 23 de mayo de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.

Hechavarría, Angela. 1997. *Las publicaciones Electrónicas: Un concepto, una clasificación y análisis de su impacto en los profesionales de la información*. Centro de Estudios y Desarrollo Profesional en Ciencias de la Información. La Habana : s.n., 1997.

Highsmith, J. Agile. 2002. *Software Development Ecosystem*. s.l. : Addison-Wesley Professional, 2002.

Inforolot. Inforolot. *Sistema de gestión documental*. [En línea] [Citado el: 17 de febrero de 2015.] <http://www.inforolot.com/index.php/es/sistema-de-gestion-documental.html>.

Ivonet, Amarilys de la C. y Hernández Martínez, Ledier. 2014. *Sistema para la edición y publicación electrónica Gaceta Oficial de la República de Cuba*. La Habana : s.n., 2014.

Joomla. 2005. Joomla. [En línea] 2005. [Citado el: 5 de febrero de 2015.] <http://www.joomla.org/about-joomla.html>.

JoomlaSpanish. 2005. JoomlaSpanish. [En línea] 2005. [Citado el: 5 de febrero de 2015.] <http://www.joomlaspanish.org/>.

Menéndez-Barzanallana, Rafael. 2006. Metodologías de desarrollo de software. [En línea] 2006. <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.

Meza, Mirna. 2011. Herramienta CASE. [En línea] abril de 2011. [Citado el: 7 de febrero de 2015.] <http://fds-herramientascase.blogspot.com/>.

Netbeans.org. 2015. Netbeans.org. *Netbeans.org*. [En línea] 2015. [Citado el: 6 de febrero de 2015.] <https://netbeans.org/features/index.html>.

Orallo, Enrique H. 2012. El Lenguaje Unificado (UML). [En línea] 2012. http://www.acta.es/medios/articulos/informatica_y_computacion/026067.pdf. 026.

Osuna, Maria Rosario y De La Cruz Gómez, Estefanía. 2010. *Los sistemas de gestión de contenido en Información y Documentación*. Salamanca : s.n., 2010.

Pedraza, Alejandro Miguel y González Olivera, Gabriel Alejandro. 2014. *Servicio de asistencia al proceso de consulta popular en la Facultad de Derecho de la Universidad de La Habana*. La Habana : s.n., 2014.

Penadés, Carmen, Letelier, Patricio y Canós, Jase. 2003. *Metodologías Ágiles en el Desarrollo de Software*. Metodologías Ágiles en el Desarrollo de Software. Alicante : s.n., 2003.

Postgresql.org. 2010. Postgresql.org. [En línea] 2010. [Citado el: 5 de diciembre de 2014.] http://www.postgresql.org.es/sobre_postgresql.

BIBLIOGRAFÍA CONSULTADA

Pressman, Roger. 2002. *Ingeniería de software*. 2002.

Priolo, Sebastián. 2007. Programación Extrema. http://www.fcad.uner.edu.ar/jai/6JAI/XP_6JAI.pdf. [En línea] 2007. [Citado el: 6 de diciembre de 2014.]

Proal Aguilar, Carlos. 2006. Modelado de datos. [En línea] 2006. [Citado el: 8 de junio de 2015.] <http://ict.udlap.mx/people/carlos/is341/bases02.html>.

registroficial.gob.ec. 2014. registroficial.gob.ec. *registroficial.gob.ec*. [En línea] 2014. [Citado el: 10 de diciembre de 2015.] <https://www.registroficial.gob.ec/>.

Rodríguez, F.G. 2012. Experto en Drupal 7. Nivel avanzado. s.l. : Forcontu, 2012.

Ruiz, Alfredo. 2014. *¿Cuáles son las características principales del HTML?* 2014.

Sarduy, Yanetsys y Urra, Pedro. 2006. *Sistemas de gestión de contenidos: En busca de una e una plataforma ideal*. 2006.

Slideshare.net. 2012. Slideshare.net. [En línea] 2012. [Citado el: 5 de diciembre de 2014.] <http://es.slideshare.net/>.

SQLite. 2015. <http://www.sqlite.org/>. [En línea] 2015. [Citado el: 7 de febrero de 2015.]

SQLITE. 2015. SQLITE. [En línea] 2015. [Citado el: 5 de febrero de 2015.] <http://www.sqlite.org>.

Tocto, Esteban . 2011. *Optimización y cuantificación utilizando BPMN*. 2011.

Trujillo , Yadelis y Fernández, Radamés. 2013. *Subsistema Mensajería SWIFT Fase II*. La Habana : s.n., 2013.

w3c.es. 2014. w3c.es. [En línea] 2014. [Citado el: 7 de febrero de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.

wordpress.org. 2015. wordpress.org. [En línea] 2015. [Citado el: 10 de febrero de 2015.] <https://wordpress.org/about/>.

Zambrano, Omar . 2014. Qué son y para qué sirven las tarjetas CRCs. *scribd.com*. [En línea] 2014. [Citado el: 5 de junio de 2015.] <http://es.scribd.com/doc/207429482/Que-son-y-para-que-sirven-las-tarjetas-CRCs-docx#scribd>.

Anexos

Anexo 1: Acta de Aceptación de la revisión de los requisitos.



Acta de Aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del desarrollo del trabajo de diploma: **Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS**, se hace entrega del artefacto que se relaciona a continuación:

- Documento Especificación de Requisitos de Software v1.0.

Entregan

Recibe

Nombre y Apellidos: Dailenis Alcántara Frómata

Nombre y Apellidos: MsC. Yarina Amoroso Fernández

Cargo: Tesisistas

Cargo: Presidenta de la Sociedad Cubana de Derecho e Informática.

Firma:

Firma:

Comentarios: El documento Especificación de Requisitos de Software v1.0 de la Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS, fue entregado y aceptado por la parte cliente.



Fecha: 27/03/2015

Figura 14: Acta de aceptación de los requisitos

Anexo 2: Acta de Liberación Interna de Productos de Software.

Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: 18/06/2015

Emitida a favor de: Tesis Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS.

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App: Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS	1.0	0	2	Evaluación dinámica Pruebas de Funcionalidad	18/06/2015


 Ing. Raúl Velázquez Álvarez
 Especialista del Grupo Calidad




 Dailenis Alcántara Frómata
 Autor

Figura 15: Acta de liberación de la aplicación web

Anexo 3: Acta de aceptación de la aplicación web por parte del cliente.



Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del desarrollo del trabajo de diploma: **Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS**, se hace constancia de la aceptación por la parte cliente de la solución que se relaciona a continuación:

- Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS.

Entregan

Recibe

Nombre y Apellidos: Dailenis Alcántara Frómata

Nombre y Apellidos: MsC. Yarina Amoroso Fernández

Cargo: Tesistas

Cargo: Presidenta de la Sociedad Cubana de Derecho e Informática.

Firma: 

Firma: 

Comentarios: La Aplicación web para la inscripción, revisión técnica y control de vigencia de disposiciones jurídicas en el MINJUS, fue aceptada por la parte cliente, luego de comprobarse por el Grupo de Calidad CEGEL, que cada una de las funcionalidades firmadas en el documento Especificación de Requisitos de Software v1.1, fueron implementadas correctamente y obtenerse el acta de liberación de la misma.



Figura 16: Acta de aceptación de la aplicación web

Anexo 4: Modelo de datos.

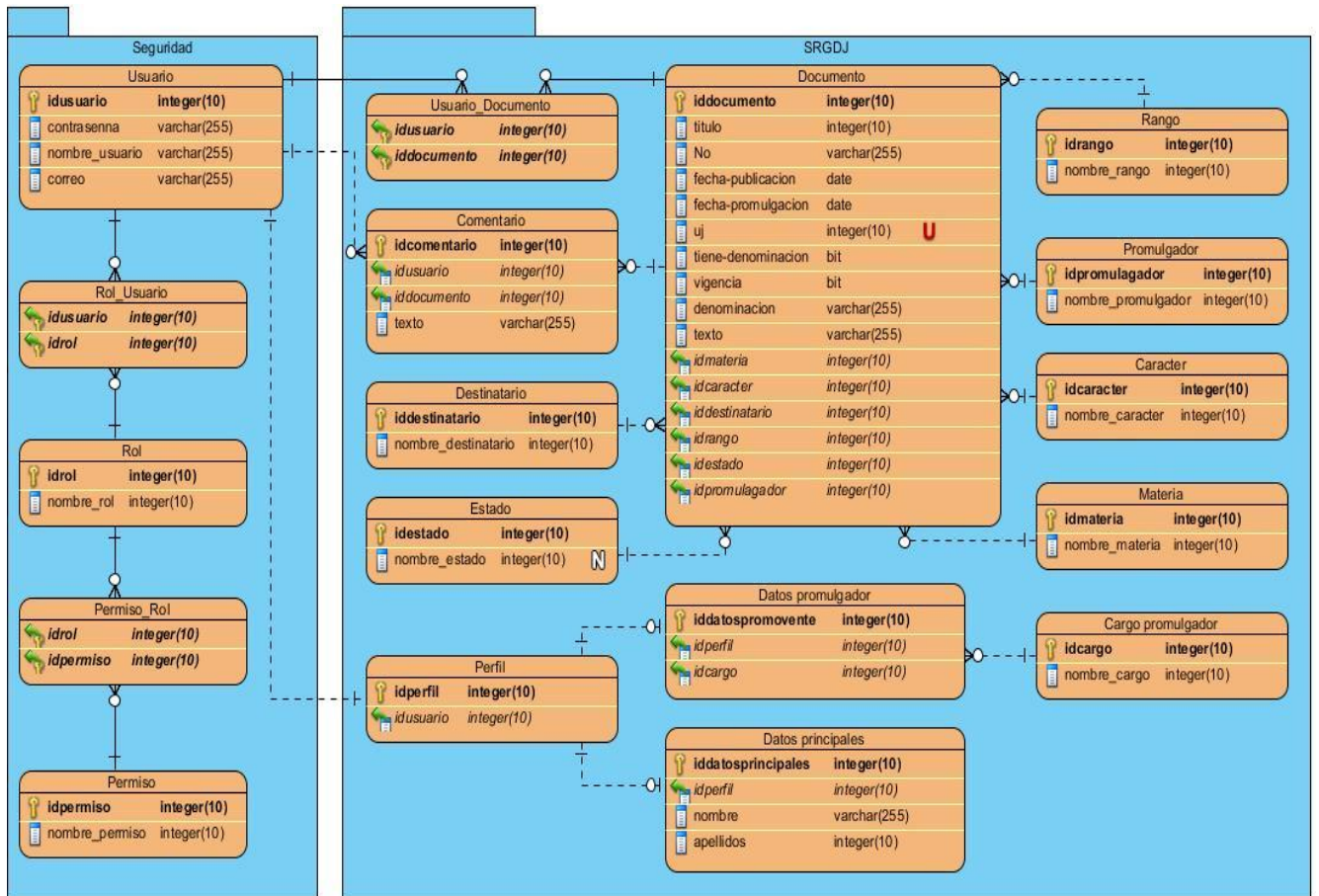


Figura 17: Modelo de datos